





DUDLEY KNOX LIBRARY  
NAVAL POSTGRADUATE SCHOOL  
MONTEREY, CALIFORNIA 93943

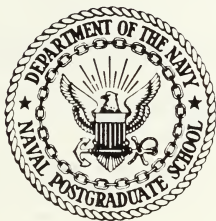






# NAVAL POSTGRADUATE SCHOOL

Monterey, California



## THESIS

CAESAR: COMMISSIONED ASSIGNMENTS EXECUTIVE SUPPORT  
FOR THE US ARMY

by

Paul A. Stipek

March 1986

Co-advisors:

T. Bui  
D. R. Dolk

Approved for public release; distribution is unlimited.

T227292





## REPORT DOCUMENTATION PAGE

REPORT SECURITY CLASSIFICATION			1b. RESTRICTIVE MARKINGS			
SECURITY CLASSIFICATION AUTHORITY			3 DISTRIBUTION/AVAILABILITY OF REPORT			
DECLASSIFICATION/DOWNGRADING SCHEDULE			Approved for public release; distribution is unlimited			
PERFORMING ORGANIZATION REPORT NUMBER(S)			5 MONITORING ORGANIZATION REPORT NUMBER(S)			
NAME OF PERFORMING ORGANIZATION		6b OFFICE SYMBOL (if applicable)	7a. NAME OF MONITORING ORGANIZATION			
Naval Postgraduate School		Code 54	Naval Postgraduate School			
ADDRESS (City, State, and ZIP Code)			7b. ADDRESS (City, State, and ZIP Code)			
Monterey, CA 93943-5000			Monterey, CA 93943-5000			
NAME OF FUNDING/SPONSORING ORGANIZATION		8b OFFICE SYMBOL (if applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER			
ADDRESS (City, State, and ZIP Code)			10 SOURCE OF FUNDING NUMBERS			
PROGRAM ELEMENT NO		PROJECT NO	TASK NO		WORK UNIT ACCESSION NO	
TITLE (Include Security Classification)						
CAESAR: COMMISSIONED ASSIGNMENTS EXECUTIVE SUPPORT FOR THE US ARMY						
PERSONAL AUTHOR(S)						
Paul A. Stipek						
TYPE OF REPORT		13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)		15 PAGE COUNT	
Master's Thesis		FROM _____ TO _____	1986 March 27		97	
SUPPLEMENTARY NOTATION						
COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)			
FIELD	GROUP	SUB-GROUP	CAESAR, DECISION SUPPORT SYSTEM, OFFICER ASSIGNMENT SYSTEM, PERSONNEL, POSITION REQUIREMENTS, PREFERENCES			
ABSTRACT (Continue on reverse if necessary and identify by block number)						
<p>The Army officer assignment system, while generally functional, is not optimal, especially with regard to consideration of officer desires and skills. It is feasible to achieve significant improvement through a decision support system that could match position requirements with officer talents and preferences. This system, when supervised by knowledgeable, involved officers, could greatly improve morale and assignment efficiency plus lower some personnel and training costs. This thesis develops a simple prototype for such a system called CAESAR. It uses data that is already available, on a database system that is substantially in place, to aid presently assigned personnel managers place the right man in the right job.</p>						
DISTRIBUTION/AVAILABILITY OF ABSTRACT			21 ABSTRACT SECURITY CLASSIFICATION			
UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS <input type="checkbox"/>			UNCLASSIFIED			
NAME OF RESPONSIBLE INDIVIDUAL			22b TELEPHONE (Include Area Code)		22c OFFICE SYMBOL	
D. R. Dolk			408 646-2260		Code 54Dk	

Approved for public release; distribution is unlimited.

CAESAR:  
Commissioned Assignments Executive Support for the US Army

by

Paul A. Stipek  
Major, United States Army  
B.S., United States Military Academy, 1973

Submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL  
March 1986

## ABSTRACT

The Army officer assignment system, while generally functional, is not optimal, especially with regard to consideration of officer desires and skills. It is feasible to achieve significant improvement through a decision support system that could match position requirements with officer talents and preferences. This system, when supervised by knowledgeable, involved officers, could greatly improve morale and assignment efficiency plus lower some personnel and training costs. This thesis develops a simple prototype for such a system called CAESAR. It uses data that is already available, on a database system that is substantially in place, to aid presently assigned personnel managers place the right man in the right job.

## THESIS DISCLAIMER

The reader is cautioned that computer program developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, extensive testing and validation are still needed. Any application of these programs without additional verification is at the risk of the user.

## TABLE OF CONTENTS

I.	INTRODUCTION . . . . .	9
A.	MOTIVATION . . . . .	9
B.	SCOPE . . . . .	10
C.	RESEARCH TECHNIQUES . . . . .	10
D.	CHAPTER AND APPENDIX SUMMARIES . . . . .	11
II.	REQUIREMENTS DETERMINATION . . . . .	12
A.	ASSIGNMENT PROCESS . . . . .	12
1.	Genus of Officer Requirements . . . . .	12
2.	Individual Officer's Role . . . . .	14
3.	Assignment Manager's Role . . . . .	20
B.	PROBLEMS . . . . .	22
1.	"Good for Your Career" . . . . .	22
2.	The Preference Statement . . . . .	27
3.	Calling/Going to Branch . . . . .	30
4.	"Needs of the Service" . . . . .	30
5.	"Right Man for the Right Job" . . . . .	31
C.	PROBLEM DEFINITION . . . . .	31
D.	ALTERNATIVES . . . . .	32
III.	CAESAR . . . . .	35
A.	GENERAL FRAMEWORK . . . . .	35
B.	DESIGN . . . . .	35
1.	Hardware . . . . .	35
2.	Data . . . . .	36
3.	Program . . . . .	39
4.	Procedures . . . . .	47
5.	Personnel . . . . .	48
IV.	CONCLUSION AND RECOMMENDATIONS . . . . .	49

A.	CONCLUSION . . . . .	49
B.	RECOMMENDATIONS . . . . .	49
1.	Implementation . . . . .	49
2.	Preference Sheet Revision . . . . .	50
3.	Officer Desires . . . . .	50
4.	CAESAR Enhancements . . . . .	51
5.	Other Assignment Systems . . . . .	53
C.	OPPORTUNITY . . . . .	53
APPENDIX A:	GLOSSARY . . . . .	55
APPENDIX B:	PROGRAM LISTING . . . . .	56
APPENDIX C:	DATA DICTIONARY . . . . .	86
APPENDIX D:	SAMPLE OUTPUT . . . . .	92
LIST OF REFERENCES	. . . . .	93
BIBLIOGRAPHY	. . . . .	95
INITIAL DISTRIBUTION LIST	. . . . .	96

## LIST OF TABLES

I	BASIC BRANCHES . . . . .	13
II	FUNCTIONAL AREAS . . . . .	14
III	DATABASE RELATIONSHIPS . . . . .	36
IV	MATCH CRITERIA . . . . .	40
V	LIST CHARACTERISTICS . . . . .	44
VI	DUTY IS PRIME FACTOR . . . . .	45
VII	LOCATION IS PRIME FACTOR . . . . .	46

## LIST OF FIGURES

2.1	DA Form 483 Mark Sense Region . . . . .	16
2.2	Comment Area . . . . .	17
2.3	Area and Duty Codes . . . . .	18
2.4	Instructions . . . . .	19
2.5	Officer's Record Brief . . . . .	21
2.6	DA Form 483 - 1975 Edition . . . . .	29
3.1	Program Flow . . . . .	41



## I. INTRODUCTION

### A. MOTIVATION

"That assignment would not be good for your career." Variations of this theme have been uttered an incalculable number of times to previously hopeful, but subsequently skeptical, Army officers in the field. The authors of these statements through the years have been the branch assignment officers at the US Army Military Personnel Center (MILPERCEN) in Alexandria, Virginia. Typically the prelude to this remark has been an optimistic expression over the telephone by an officer in the field as to what he would like his next duty station to be. A common reaction to the personnel manager's quote is one of frustration, suspicion, or contempt:

- "I don't know why he/they/the Army won't let me go there, since I'm qualified."
- "I'll bet he thinks there is something I am trying to avoid or some way I am trying to beat the system."
- "Those guys at branch don't think about us at all. All they care about is filling a position and passing the action to someone else."

Thus an adversary relationship sometimes exists between officers in the field and their assignment specialists. In their calmer, more reflective moments, most officers realize that their brothers at MILPERCEN try to do as thorough a job any officer does, constrained by time and directives. Yet the result is often unsatisfying for both the moving officer, who does not believe he is being assigned the best job available, and the branch specialist, who feels that his efforts to put the right man in the right job are unappreciated. The relationship between MILPERCEN and the officer corps does not have to be this strained. This thesis proposes a prototype computer aid to ameliorate this situation.

## B. SCOPE

The main thrust of this thesis is to demonstrate both the need and the potential for greater automation of the assignment process for commissioned Army officers through a decision support system (DSS). Keen and Wagner define a DSS as:

a computer-based system . . . which is used personally on an ongoing basis by managers and their immediate staffs in direct support of managerial activities--that is, decisions. Another term for DSS might be "executive mind-support system." [Ref. 1: p. 117]

The prototype DSS system proposed here attempts to better the performance of MILPERCEN assignment managers in the domain of matching officer skills and preferences to position requirements. The successful application of such a system could lower training costs by reducing the need for preassignment schooling. It could improve morale and reduce attrition by elevating the role of officer preferences in the assignment process. No attempt is made to exactly detail a MILPERCEN implementation, since the goal of this effort is to demonstrate possibilities, not provide a detailed architecture. Although it is the author's contention that similar systems could be developed to automate warrant officer and enlisted assignments, as well as the detailing procedures of other services, these topics will not be examined in this thesis, as each has its own problems and represent potential theses for future master's candidates.

## C. RESEARCH TECHNIQUES

The requirements determination portions of this work are based primarily on the author's observations of, and experience with, the assignment process in action during his nearly 13 years as an Army officer. Face-to-face and telephonic interviews with assignment personnel and affected

officers were also central to this effort. In order to encourage candor from those interviewed, these conversations were generally conducted under the premise that they were not for attribution. This research pattern naturally leads to a limited use of referenced sources, but enhances the relevance of the product.

#### D. CHAPTER AND APPENDIX SUMMARIES

This thesis derives its organization from a variation of the system development steps outlined by Kroenke [Ref. 2]. Chapter II demonstrates the requirement for computer assistance by explaining part of the current officer personnel management process. The emphasis is on how that routine is perceived by officers in the field. Chapter III discusses the design of the prototype, Commissioned Assignments Executive Support for the US Army (CAESAR). Chapter IV summarizes the findings of the thesis and lists the author's recommendations for implementation of such a system, further study and corrective actions in the assignment process.

Appendix A contains a glossary of acronyms used in the main body of the thesis. Appendix B shows the program listing. Appendix C is an abbreviated data dictionary for the program. Appendix D has an example of typical output.

## II. REQUIREMENTS DETERMINATION

### A. ASSIGNMENT PROCESS

#### 1. Genus of Officer Requirements

Generally, each unit/organization in the Army has a document that authorizes the personnel and equipment to make up the unit. Typically this document is called either a Modification Table of Organization and Equipment (MTOE) for units that can be deployed to war, or a Table of Distribution and Allowances (TDA), for those organizations that do not deploy. These publications form a significant portion of the Army Authorization Document System (TAADS), which is a large database of organizational information. These documents contain a nine-digit code, called a Position Requirement Code (PRC), for each required officer position listed [Ref. 3: pp. 3-4]. This code specifies the skills the officer holding this position should have. The MTOE earns its first name because its parent, the Table of Organization and Equipment (TOE), represents theoretical wartime requirements which are reduced in the MTOE. These lesser amounts are tagged "authorized" and are usually due to resource constraints or the reduced peacetime needs of the unit. The "authorized" level is the maximum figure that the unit personnel officer can requisition for his unit, as vacancies are projected due to losses or organizational changes. In the Army, there are about 63,000 authorized requirements for basic branch commissioned officers from second lieutenant through colonel scattered throughout the world [Ref. 4]. The basic branches are divided into combat arms, combat support arms, and combat service support as shown in Table I.

The local personnel managers send these requirements up their chain of command until they reach MILPERCEN. There

TABLE I  
BASIC BRANCHES

	<u>Branch</u>	<u>Specialty Code (SC)</u>
Combat Arms	Infantry	11
	Armor	12
	Field Artillery	13
	Air Defense Artillery	14
	Aviation	15
	Corps of Engineers	21
Combat Support Arms	Signal Corps	25
	Military Police	31
	Military Intelligence	35
	Chemical Corps	74
Combat Service Support	Adjutant General Corps	42
	Finance Corps	44
	Ordnance Corps	91
	Quartermaster Corps	92
	Transportation Corps	95

each requirement must be validated by the Distribution Division. This office manages the Officer Distribution Plan (ODP), a program that matches the constrained officer inventory to the more numerous list of officer requirements. If filling the request under consideration will not place the requesting command over its ODP limit, Distribution Division forwards it to the assignment branch designated to fill that requirement. [Ref. 5: p. 12] This branch may have been chosen because the requirement is directly related to that branch, such as an infantry or aviation assignment, or because it is that branch's turn to provide someone with a more general, branch-immaterial functional area skill, such as those found in Table II.

MILPERCEN's routine is to begin processing an overseas officer request nine months before the projected reporting date of the officer, six months for a Continental United States (CONUS) move. The branch goal is to fill the requirement within 30 days, thus giving the inbound officer

TABLE II  
FUNCTIONAL AREAS

<u>Functional Area</u>	<u>SC</u>
Special Operations	18
Personnel Management	41
Comptroller	45
Public Affairs	46
Foreign Area Operations	48
Research and Development	51
Nuclear Weapons	52
Systems Automation Officer	53
Operations, Plans, Training	54
Procurement	97

five to eight month's notice. To further control the process, CONUS assignments are processed during odd-numbered months and overseas requisitions are worked in alternate months. [Ref. 5: p. 12] Short notice, high priority, or difficult to fill assignments frequently upset this routine, however.

## 2. Individual Officer's Role

Officers are frequently told that they are the primary managers of their own careers. They are expected to keep abreast of officer management issues and to consult with superiors, branch personnel specialists, and official and unofficial publications as to career development. They are also told that each job they are assigned is important, or else the Army would not expend its limited personnel assets on it. Therefore all duty assignments should be executed to the best of their ability [Ref. 6]. This is in marked contrast to the "ticket-punching" mentality of the 1960's and 70's [Ref. 7: p. 10], which viewed all other assignments as holding patterns between command and professional development schooling postings. The assignment process is considered to be part of the Officer Personnel

Management System (OPMS). While personal career preferences are clearly lower in priority to needs of the service in OPMS, officers are regularly encouraged to make their preferences known to assignment officers [Ref. 8: p. 5].

The official mechanism for accomplishing this task is use of the Officer Assignment Preference Statement, Department of the Army (DA) Form 483. See Figures 2.1 thru 2.4.

The current version, implemented in early 1985, is a four-page document which includes:

- a. mark sense positions to indicate assignment preferences, schooling desires, and personal data,
- b. address and comment areas,
- c. a list of the codes to be used in the mark sense portion, and
- d. instructions.

This form:

allows officers to express their assignment and duty preferences. Individual preferences are considered by assignment managers each time an officer is reassigned by (MILPERCEN). Every effort is made to comply with the officer's preferences consistent with the needs of the Army. [Ref. 9: pp. 3-4]

Officers fill out the form with a #2 pencil and mail it directly to their branch at MILPERCEN without any intermediate review. There the "mark sense data on the first page of DA Form 483 will be stored on the automated Officer Master File (OMF) maintained at MILPERCEN." [Ref. 9: p. 4] This information is available to the assignment officer via a terminal in the office, manned by a technician or the assignment officer himself.

Individuals can also maintain personal contact with assignment executives by either visiting MILPERCEN or staying in contact by phone [Ref. 5: p. 28]. Though many a finger, worn down in search of an open telephone line to branch, may question its practicality, phone calls to

SOCIAL SECURITY NUMBER		CURRENT DATE		I AM A:		OFFICER ASSIGNMENT PREFERENCE STATEMENT			
<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">JAN</div> <div style="border: 1px solid black; padding: 2px;">FEB</div> <div style="border: 1px solid black; padding: 2px;">MAR</div> <div style="border: 1px solid black; padding: 2px;">APR</div> <div style="border: 1px solid black; padding: 2px;">MAY</div> <div style="border: 1px solid black; padding: 2px;">JUN</div> <div style="border: 1px solid black; padding: 2px;">JUL</div> <div style="border: 1px solid black; padding: 2px;">AUG</div> <div style="border: 1px solid black; padding: 2px;">SEP</div> <div style="border: 1px solid black; padding: 2px;">OCT</div> <div style="border: 1px solid black; padding: 2px;">NOV</div> <div style="border: 1px solid black; padding: 2px;">DEC</div> </div> <div> <div style="border: 1px solid black; padding: 2px;">YEAR</div> <div style="border: 1px solid black; padding: 2px;">MONTH</div> <div style="border: 1px solid black; padding: 2px;">DAY</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">COMMISSIONED</div> <div style="border: 1px solid black; padding: 2px;">OFFICER</div> </div> <div> <div style="border: 1px solid black; padding: 2px;">WARRANT</div> <div style="border: 1px solid black; padding: 2px;">OFFICER</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">USE NO. 2 PENCIL ONLY</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">DO NOT FOLD, TEAR OR STAPLE</div> </div> </div>	
<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>	
<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>			
<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>			
<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>			
<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>			
<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>			
<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>			
<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>			
<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>			
<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>			
<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">4</div> <div style="border: 1px solid black; padding: 2px;">5</div> <div style="border: 1px solid black; padding: 2px;">6</div> <div style="border: 1px solid black; padding: 2px;">7</div> <div style="border: 1px solid black; padding: 2px;">8</div> <div style="border: 1px solid black; padding: 2px;">9</div> <div style="border: 1px solid black; padding: 2px;">0</div> </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <div style="border: 1px solid black; padding: 2px;">1</div> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">3</div> </div></div>							



## This image shows a single sheet of white paper with horizontal blue or grey ruling lines, typical of notebook paper. The lines are evenly spaced and run across the width of the page. There is no handwriting or other markings on the paper.

NAME	RANK	BRANCH (SSI FOR AMEDD OFFICERS)

Figure 2.2 Comment Area.

# PREFERENCE CODES

## GUNNY PREFERENCES

Alaska	01	Idaho	16	Massachusetts	28	Rhode Island	44	Q2 - ALASKA	OM - NORTH AMERICA
Alaska	AN	Idaho	17	Massachusetts	29	Rhode Island	45	Q3 - FEDERAL REPUBLIC OF GERMANY	OM - NORTH AMERICA
Alaska	AN	Idaho	18	Massachusetts	30	Providence	46	Q4 - HAWAII	OM - NORTH AMERICA
Alaska	AN	Idaho	19	Massachusetts	31	Providence	47	Q5 - JAPAN	OM - NORTH AMERICA
Alaska	AN	Idaho	20	Massachusetts	32	Providence	48	Q6 - REPUBLIC OF KOREA	OM - NORTH AMERICA
Alaska	AN	Idaho	21	Massachusetts	33	Providence	49	Q7 - PANAMA	OM - NORTH AMERICA
Alaska	AN	Idaho	22	Massachusetts	34	Providence	50	Q8 - SUB SAHARA AFRICA	OM - NORTH AMERICA
Alaska	AN	Idaho	23	Massachusetts	35	Providence	51	Q9 - CANADIAN	OM - NORTH AMERICA
Alaska	AN	Idaho	24	Massachusetts	36	Providence	52	Q10 - GUINIA	OM - NORTH AMERICA
Alaska	AN	Idaho	25	Massachusetts	37	Providence	53	Q11 - LIBERIA	OM - NORTH AMERICA
Alaska	AN	Idaho	26	Massachusetts	38	Providence	54	Q12 - MALAWI	OM - NORTH AMERICA
Alaska	AN	Idaho	27	Massachusetts	39	Providence	55	Q13 - SOUTH AFRICA	OM - NORTH AMERICA
Alaska	AN	Idaho	28	Massachusetts	40	Providence	56	Q14 - CHINA (PEK)	OM - NORTH AMERICA
Alaska	AN	Idaho	29	Massachusetts	41	Providence	57	Q15 - CHINA (TIANJIN)	OM - NORTH AMERICA
Alaska	AN	Idaho	30	Massachusetts	42	Providence	58	Q16 - CHINA (HONGKONG)	OM - NORTH AMERICA
Alaska	AN	Idaho	31	Massachusetts	43	Providence	59	Q17 - BANGLADESH	OM - NORTH AMERICA
Alaska	AN	Idaho	32	Massachusetts	44	Providence	60	Q18 - BURUNDI	OM - NORTH AMERICA
Alaska	AN	Idaho	33	Massachusetts	45	Providence	61	Q19 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	34	Massachusetts	46	Providence	62	Q20 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	35	Massachusetts	47	Providence	63	Q21 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	36	Massachusetts	48	Providence	64	Q22 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	37	Massachusetts	49	Providence	65	Q23 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	38	Massachusetts	50	Providence	66	Q24 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	39	Massachusetts	51	Providence	67	Q25 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	40	Massachusetts	52	Providence	68	Q26 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	41	Massachusetts	53	Providence	69	Q27 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	42	Massachusetts	54	Providence	70	Q28 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	43	Massachusetts	55	Providence	71	Q29 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	44	Massachusetts	56	Providence	72	Q30 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	45	Massachusetts	57	Providence	73	Q31 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	46	Massachusetts	58	Providence	74	Q32 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	47	Massachusetts	59	Providence	75	Q33 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	48	Massachusetts	60	Providence	76	Q34 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	49	Massachusetts	61	Providence	77	Q35 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	50	Massachusetts	62	Providence	78	Q36 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	51	Massachusetts	63	Providence	79	Q37 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	52	Massachusetts	64	Providence	80	Q38 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	53	Massachusetts	65	Providence	81	Q39 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	54	Massachusetts	66	Providence	82	Q40 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	55	Massachusetts	67	Providence	83	Q41 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	56	Massachusetts	68	Providence	84	Q42 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	57	Massachusetts	69	Providence	85	Q43 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	58	Massachusetts	70	Providence	86	Q44 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	59	Massachusetts	71	Providence	87	Q45 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	60	Massachusetts	72	Providence	88	Q46 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	61	Massachusetts	73	Providence	89	Q47 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	62	Massachusetts	74	Providence	90	Q48 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	63	Massachusetts	75	Providence	91	Q49 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	64	Massachusetts	76	Providence	92	Q50 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	65	Massachusetts	77	Providence	93	Q51 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	66	Massachusetts	78	Providence	94	Q52 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	67	Massachusetts	79	Providence	95	Q53 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	68	Massachusetts	80	Providence	96	Q54 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	69	Massachusetts	81	Providence	97	Q55 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	70	Massachusetts	82	Providence	98	Q56 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	71	Massachusetts	83	Providence	99	Q57 - CAMBODIA	OM - NORTH AMERICA
Alaska	AN	Idaho	72	Massachusetts	84	Providence	100	Q58 - CAMBODIA	OM - NORTH AMERICA

## DUTY PREFERENCES

A	Command	L	Instructor Service School
B	Joint and Combined Staff	M	Instructor USA
C	DA Staff	N	Undergraduate Degree Completion
D	MAJCOM Staff	O	Advanced Civil Schooling
E	Staff - Troop (Division Installation and Below)	P	Specialty Training
F	Field Operating Agencies	Q	Other
G	Recruiting Command	R	No Preference
H	Reserve Component	S	Special Operations
I	Service School	T	Research (AMEDG only)
J	Instructor	U	Field Duty Administration (AMEDG only)
K	Instructor ROTC	V	Field Facility (AMEDG only)

## RESERVE COMPONENT SUPPORT DESIGNS

RB	First ROTC Region	- Ft Bragg
RX	Second ROTC Region	- Ft Knox
RR	Third ROTC Region	- Ft Riley
RL	Fourth ROTC Region	- Ft Lewis
AF	Fifth ROTC Region	- Ft Meade
AJ	Sixth ROTC Region	- Ft Gilman
AL	Seventh ROTC Region	- Ft McPherson
AM	Eighth ROTC Region	- Ft Seward
AN	Ninth ROTC Region	- Ft Sam Houston
AO	Tenth ROTC Region	- Presidio of San Francisco

Figure 2.3 Area and Duty Codes.

## INSTRUCTIONS

CURRENT DATE - Give current month and year. This will be the date printed on your ORB in the remarks section to show you that your Preference Statement has been received.

### ASSIGNMENT PREFERENCE -

CONUS - Enter 3 CONUS location choice preferences from attached codes.

OVERSEAS LONG TOUR<sup>1</sup> - Enter 2 overseas location preferences from attached codes.

OVERSEAS SHORT TOUR<sup>1</sup> - Enter 2 overseas location preferences from attached codes.

<sup>1</sup>See AR 614-30 for long and short tour areas.

DUTY PREFERENCE - Enter 3 preferences from attached codes.

LAST NAME - Enter first 5 characters of last name. For very short names, skip a space after last name and enter as many positions of first name as possible.

GENERAL - Specifically comment on any item which you feel will clarify any point which you want considered in your next assignment. Use continuation sheet to make comments if required. When completed, this preference statement should be sent to one of the following addresses, as appropriate. Do not fold any part of this form, mail in 9 inch by 12 inch envelope.

JAG - The Judge Advocate General  
ATTN: DAJA-PT  
Department of the Army  
Washington, DC 20310

Chaplain - Chief of Chaplains  
ATTN: DACH-REA  
Department of the Army  
Washington, DC 20310

Medical - US Army Medical Department Personnel Support Agency  
ATTN: SGPE ZB  
1900 Half Street, SW  
Washington, DC 20324

All Other Officers - US Army Military Personnel Center  
ATTN: DAPC-iAppropriate Branch for 01-05)  
ATTN: DAPC-OPC (For Colonels)  
ATTN: DAPC-OPW (For Warrant Officers)  
200 Stovall Street  
Alexandria, VA 22332

## FAMILY CONSIDERATIONS

### EXCEPTIONAL FAMILY MEMBER (EFM)

If you have an exceptional family member (one who requires special medical and/or educational treatment and/or facilities), complete the following information.

- Is your exceptional family member enrolled in the EFM program?

YES \_\_\_\_\_ NO \_\_\_\_\_

- What is the age of your exceptional family member?

- Briefly describe your exceptional family member's condition:

---

---

---

---

To have your exceptional family member situation fully considered, formal enrollment in the program is necessary. Refer to AR 614-203 and your servicing Unit/Bn PAC or MILPO.

Figure 2.4 Instructions.

MILPERCEN by all officers is encouraged by official policy [Ref. 10: p. 11].

### 3. Assignment Manager's Role

The branch personnel manager receives the routine requirements each month in the form of a computer printout. It contains the new requirements of the current CONUS or overseas assignment cycle plus whatever requirements may not have been filled from the prior month.

Each branch officer

focuses on a specific population of officers holding the same grade and specialty. This means that within the smaller branches and specialties, officers of the same grade are managed by a single assignment officer. Within the larger branches, such as Infantry, graded populations are broken down into a workable size and managed by several assignment officers.

[Ref. 5: pp. 1,12]

Each assignment executive operates by his own method at this point. Some keep drawers full of files ordered by when officers moved last. Those who have not moved for a long time are on top and are the first considered when a new requirement comes in. [Ref. 5: p. 28] Other managers keep books of Officer Record Briefs (ORB), one page resumes of officers' careers, replete with codes used in PRC's (Figure 2.5) [Ref. 11]. These are used to provide snapshots of officers to determine if they should be considered when new requirements cross the manager's desk. Still others use their assistants, called technicians, or newly operational computer terminals, to query the OMF to determine who is available to be reassigned. These deskside terminals also make it possible to examine the preference statements of those under consideration for reassignment to try to match desires with qualifications [Ref. 8: p. 5].

Once a potential match has been found, most assignment officers will make some attempt to contact the nominee

[illegible]

Figure 2.5 Officer's Record Brief.

for input into the process. For some of the most routine assignments, such as:

- operational pilot assignments after flight school,
- officer advanced course attendance after an initial CONUS tour, or
- orders to Command and General Staff College after selection by the board,

less time is spent in this interaction, since choices are both obvious and limited. From the output list, he picks the best qualified based on his current operating guidelines and personal judgment, runs the selection through the branch review and approval system, notifies the losing commander of his intent to move the officer, and awaits any strenuous objection from the command. If no problems develop, he initiates a request for orders.

The Army must have officers to fill all the authorized jobs. Some positions are highly desirable assignments and are easy to fill. Others are highly undesirable and more difficult to find volunteers for. Personnel managers frequently remind the officer corps that the needs of the Army come first. Therefore, inevitably, some people will be assigned to jobs they do not like or want. This can produce an adversary relationship between officers in the field and their assignment specialists in MILPERCEN. It seems that much of this tension is unnecessary. With so many positions available, it seems highly unlikely that, given the right tools, one could not find a job for almost every officer that at least generally fits his preferences and matches his skills.

## B. PROBLEMS

### 1. "Good for Your Career"

The assignment officer's subjective evaluation of what is "good" for an officer's career, which is frequently promulgated during the branch telephone calls or interviews is a major source of annoyance. It is generally accepted by

the officer corps that there are certain "mandatory" assignments, such as branch advanced courses and utilization tours after flight and graduate schools. However, whatever else is "good" for one's career seems to vary from assignment officer to assignment officer and is further complicated by shifting personnel philosophies hatched by changes in branch, division, MILPERCEN, and Army chiefs, as well as a migrating officer personnel management system [Ref. 12]. Thus what is "good" one year might be a career risk the next. Career development is ranked by personnel managers below the specific current needs of the Army (though the two are linked by some notion that the Army in general needs officers whose careers have developed "correctly") and well above individual desires [Ref. 9: p. 3]. This dimension leads to assignment patterns that frequently leave officers bewildered and frustrated. Many officers feel that assignment officer career advice has not been all that inspired over the years. These officers feel that they, as individuals, should have maximum latitude over their own career development. After all, it is the individual, not the assignment executive, who suffers the impact of an improperly nurtured career. The paternalistic attitude that "MILPERCEN knows best" is often taken to task.

a. Army Aviator's Saga

The career management history of Army aviators provides an example of shifting "goodness" policies. With the creation of the Department of the Air Force in 1947, aviation in the Army moved from a full time career corps, or branch, to a part time special skill possessed by relatively few Army officers, all of whom were members of other branches, usually in the combat arms. As the helicopter became important, more and more officers became pilots, but it was still quite clear, especially in the combat arms, that the road to success was generally detoured by aviation

assignments. It was useful (and profitable due to flight pay and flight school per diem) to spend a tour in aviation to broaden professional development by learning about that aspect of the Army. However, promotions were to be earned in one's branch, especially by assuming company command as a captain. Those who took repetitive tours in aviation had very dismal promotion outlooks. As the Vietnam war peaked, however, due to the large number of aviation units, one-year tours, and relatively high casualties, it became clear that many pilots would be required to serve multiple aviation tours. It was common for pilots to have two, even three, combat aviation tours. Aviation companies, because of their expense and complexity, had majors as company commanders. This created a dilemma for aviation captains. Their service needed them in combat as pilots, so many did not have time to go back to their branches to be line company commanders, which they knew could be devastating to their promotion potential. In recognition of this fact, a letter from a four-star general was inserted in many combat aviators' files to inform future promotion boards that the officer in question had been required to deviate from the normal career pattern through no fault of his own. However, in the postwar reductions in force, both overt and through promotion passovers, Vietnam-era aviators fared very badly, in spite of having been told how combat tours would be "good" for them.

With this example in mind, Army aviators in the 1970's were careful to spend the required time in their "carrier" branches [Ref. 13]. Late in that decade, however, it became clear to the Army leadership that the projected shortage of company-grade (lieutenants and captains) aviators, the expense of modern helicopters, and the complexity of survival in the emerging high threat air defense cried for a corps of professional aviators rather than a part-time



force [Ref. 14]. Thus aviation was elevated from a skill to a specialty, though its creation as a branch was still controversial. Once again aviators were being told that it was no longer necessary for them to command infantry companies or artillery batteries, even though they still wore that branch insignia [Ref. 15]. An "Aviation Management Branch" was created in MILPERCEN to handle aviator assignments. It had most of the functions of a combat arms branch without officially being one, due to remaining institutional fears that the Army Air Corps/U.S. Air Force experience might be repeated. Aviators were once again told that command as a captain was no longer required since they would get aviation companies as majors. Finally, in April 1983, after some uneven promotion results, Aviation was given full branch status.

It was commonly believed by aviation captains that one of the prime motivations to create the new Aviation branch was to formalize the different career pattern for aviators. They were to spend their initial years flying, go to the appropriate schools, develop their alternate specialties, and then return to aviation as majors for command. Many post-Vietnam era aviators, in coordination with branch assignment officers, launched themselves on this career path. In the mid-1980's, the deck was shuffled once again. Aviation branch from its inception had been designated a combat arms branch, even though many of its units are involved in combat support and combat service support functions. It had this variant career pattern that separated it from the other combat arms. So in an effort to simplify aviation units, to separate combat functions from support, to elevate the level of aviation commands, to provide more opportunity for command, and to emulate standard combat arms career patterns and organizations, aviation began to restructure. Platoons, formerly led by captains, became

companies commanded by captains. Similarly companies became battalions and old battalions formed the bulk of new aviation brigades.

Thus one of the reasons for Aviation branch's existence was eliminated after the branch was formulated. While the overall value of this restructuring remains to be proven, some of its casualties are those year-groups of officers who were captains when aviation commanders were majors, went to non-flying jobs, and are now majors when the commanders became captains. The concern of these officers who did things that were "good" for their careers is they will be non-competitive for promotion to lieutenant colonel as combat arms officers without experience as company commanders.

b. The Advanced Course

One would think that a branch advanced course, a six to nine month school for captains to hone branch and staff skills, would represent a great opportunity for both assignment officers and students. Here scores, even hundreds, of officers of equivalent experience in a given branch are graduating on the same day. Thus, barring extremely esoteric requests, like Army liaison to Australia or aviation advisor in Thailand, it should be relatively easy to honor individual preferences in assignments for such a relatively interchangeable group. Yet experience indicates that officers are frequently disappointed by their postings after advanced course graduation. In a 1977 Infantry Officer Advanced Course, the branch chief told the assembled students that Infantry branch (before the existence of the current Aviation branch) badly needed helicopter pilots and Special Forces (SF) team leaders. He encouraged all who were physically qualified to apply for flight training and the others to consider volunteering for SF. (It is interesting to note that in the previous Army

reduction in force, large numbers of those released were aviators or SF-qualified.) Yet in an assignment interview two days later, an officer with a valid flight physical on file was told, upon requesting flight school, that it would be bad for his career. The bimonthly assignment cycle discussed earlier represents a common refuge for personnel managers who are trying to explain why, in a given month, they may tell one officer he cannot have a certain job and then give that exact job to his acquaintance a few weeks later, when the next cycle of requisitions are processed [Ref. 5: p. 28]. For this class, the cycle problem was minimal since requirements both in CONUS and out were available. Nevertheless, some students who had come from Germany and wanted to return were told they could not ("bad for your career") while others were given orders to Germany, though they had expressed a preference to remain in CONUS. On one occasion two such officers went to an interview together, asking that their assignments be switched between each other. Common graduation notwithstanding, their request was disapproved. Some posts, such as Fort Bragg, North Carolina, and Fort Hood, Texas, are anathema to many officers, yet others who request repeated assignments to these places are chastised by managers for "homesteading," which, of course, is not "good for your career."

## 2. The Preference Statement

One of the two most common methods of determining an officer's desires, is the previously discussed preference statement, DA Form 483, nicknamed the "Dream Sheet." This working document contains coded assignment preference data. Its heart is the "Assignment Preference" section in which the officer can communicate to his branch seven locations, three in CONUS and four overseas, and three choices of duty he would like during three types of tours: CONUS, overseas accompanied (long--usually three years), and overseas

unaccompanied (short--usually one year). On its face, this form is a good mechanism for helping direct the assignment officer toward billets of one's desire.

The previous, manual edition of this form (Figure 2.6) was more comprehensive than the current edition. It allowed nine locations to be selected, permitted differentiation between preferences for long and short tours, and enabled the officer to selectively indicate whether duty or location was his prime concern in his preference for each of the three types of tours.

Despite frequent assurances to the contrary [Ref. 5: p. 28], and warnings about the result of failing to submit one [Ref. 8: p. 5], an abiding, unshakeable belief exists in some parts of the officer corps that these Form 483's are simply another item on a personnel records inventory checklist and are not read at all. An item of corollary evidence to this theory occurred when, in 1981, the Army proposed to automate the 483 so that preference data could be in the OMF data base. Initially, the personnel officials claimed that they had limited data storage capacity and thus could store only ten items. MILPERCEN chose to store nine of the 18 assignment and duty possibilities listed on the original DA Form 483 and the date of the last preference statement. [Ref. 16] The main purpose of the 483 date was to determine the currency of the form from a monitor point of view. That memory space could better have been used to store another job option, if attempting to make managers aware of individual desires was the overriding purpose of the form. This automated system was never fully implemented due to initial difficulties in keying the information into the data base and resistance on the part of assignment officers.

Many officers in the field still believe that their preferences for their next assignment are ignored. By the end of 1985, less than 20% of Army officers had updated their preferences with the new form [Ref. 8: p. 5]. Even



MILPERCEN specialists sometimes admit that they never believed that 483's were worthwhile before their current assignment. There is still some resistance to the automated preference statement at MILPERCEN. Assignment officers complain that the screen printout of the preference data is in code rather than in text, so that as much time is now spent looking up location codes as was used previously in reading the manual preference statements. In fact, one manager recommended that a submitting officer write a summary paragraph of preference data in the remarks section to ensure that the assignment officer understood what the preference statement was supposed to say. Another comment was that the most useful thing about the mark sense preference statement was the current phone number for the submitting officer it provided.

### 3. Calling/Going to Branch

The second method, tried and true, is to call or visit MILPERCEN and attempt to communicate one's desires. This process seems to work:

- if the assignment officer is contacted at the right time (not before he is looking at the individual for reassignment and not after he has initiated action to cut orders,)
- if the calling officer asks for something the manager has available at that time for fill, and
- if the caller does not try to "hurt his career."

### 4. "Needs of the Service"

There is widespread dissatisfaction with the results of individual participation in the process. This attitude is traditionally answered by a reminder that the needs of the service outweigh individual preferences. However, the point can be made that the needs of the Army are best served by officers who are motivated by the knowledge that:

- they made their own informed decision on a career pattern,
- they determined their own preference for assignments,
- and were given those positions, when reasonably available, by a supportive branch assignment officer, trying to honor those career choices.

## 5. "Right Man for the Right Job"

A final difficulty is the matching of skills and training to job assignments. Army officers receive a variety of schooling: branch, general flight, aircraft-specific, parachute, SF, language, and so forth. Most of these courses have an associated skill code entered into personnel records, identifying officers so trained. There are dozens of these codes that an officer can accrue in thousands of combinations [Ref. 3: pp. 53-70]. The assignment process generally does a good job in matching skills at a macro level. For example, it usually assigns infantry officers to infantry jobs and sends pilots to aviation positions. It does not align special skills very well. For example, at Fort Bragg in 1980, there were two positions for SF-qualified aviator captains (a rare combination of skills for the reasons stated earlier.) Yet, though such personnel were on the post, the jobs were filled by non-SF aviators, a major and a lieutenant. This fact was understandable when one realized that although the additional skill codes were contained in the OMF, present on the authorization manning documents, and available to assignment executives, they were not tracked in the assignment process.

MILPERCEN officials have recently begun using some automated interface between personnel databases and the assignment selection process, such as the OMF query system previously mentioned and the newly automated Married Army Couples (MAC) program [Ref. 17]. There is also a developing awareness that more automation improvements can be achieved in areas such as the enlisted [Ref. 18] and general officer [Ref. 19] assignment systems.

### C. PROBLEM DEFINITION

A common perception is that three things often seem to be absent in assignment officer actions:

1. an appreciation for the currently popular, though commonly lip-serviced, idea that "every job is a good

job" [Ref. 7: p. 10] and deserves to be done well for the good of the Army,

2. an understanding that people tend to perform well and succeed in jobs they like, have received formal schooling in, or had a role in choosing for themselves and, conversely, to do poorly in other types of positions, plus
3. an internalization of the concept that each officer is supposed to bear ultimate responsibility for his own career management.

The sheer complexity of trying to match rank, branch, skill, special training, and preference to Army requirements for all 63,000 officers is hopelessly beyond the unassisted mental capacity of any group of personnel managers. The problem is how to optimize the assignment process to juggle the needs of the service both in jobs and tour length, proper career management, skill training, and officer preferences and motivation, to attempt to put the right man in the right position at the right time.

#### D. ALTERNATIVES

Several options exist. The simplest is to do nothing. In an overall sense, the current system does work. One way or another, officers are found to fulfill the needs of the Army. However, the feeling of being a cog in the "Green Machine," reinforced by the relatively low esteem which officer desire seems to enjoy in the assignment process, tends to lower officer morale. It has been a cause for early retirement and resignation, with the attendant costs of training replacements. Also the current system leads to politics in the process which wastes time and ties up assignment managers and their telephones. It leads to additional training costs since, if a properly trained officer is not assigned, the present officer must be sent to school. Thus a better system should be found.

A second alternative is to expand the assignment officer work force, giving each officer less of a clientele to work with, enabling each to know their officers' skills, needs,



and preferences in greater detail. Theoretically this could work, but the personnel drain on the rest of the Army to dramatically boost MILPERCEN strength would be significant. As the Army moves to increase combat strength by filling new divisions with the personnel spaces saved by leaner support services, it is unlikely that such a personnel increase would be favorably received. Also, a proliferation of managers would naturally cause further dilution of assignment and career policy standardization by an even greater number of interpretations. More extensive telecommunications systems would need to be installed and more families would be exposed to the financial hardship of duty in Washington D.C. Thus this alternative seems costly and of doubtful practicality.

The third choice is a computer solution. A prototype DSS could be developed to demonstrate the validity of a computer-aided assignment process. By using the already computerized requirements data, employing the existing OMF resources, and expanding the automation of the Form 483 by directly tying the preference statement to the decision-making process, this DSS would enhance the role of the individual officer and aid the assignment manager by matching requirements to skills and desires to provide a list of nominees for each position. A working prototype should be relatively easy to fully develop and implement. This database system should improve the assignment process with little or no additional personnel and equipment costs, since the operators and maintainers could be the presently assigned MILPERCEN staff and the OMF is already a fully operational database system. Since the process to be automated is more time-consuming than complex, a standard database language should suffice, easing rapid program development. Computer software and, perhaps, some computer hardware investment will be required, but after the initial

development and implementation period is over, sustainment costs should be low. By elevating the value of the preferences of officers in the field, it could reduce attrition, lower training costs, and cause a concurrent rise in officer morale and performance. Therefore, with computer help, a more satisfactory solution to the assignment dilemma appears feasible from technical, schedule, and cost viewpoints.

### III. CAESAR

#### A. GENERAL FRAMEWORK

CAESAR is a program primarily concerned with matching the job requirements of the PRC with the officer skills found in the OMF. It also assesses the relative priority of the projected assignment in comparison with the desires of the individual officer as expressed in the officer's preference statement. While these actions are not particularly complex for a database computer system, the number of possible combinations make it impractical for the human assignment manager to consider them all. So he is often forced to consider only the most important skill requirements, leaving additional skill and preference information behind. The CAESAR prototype meets the definition of a DSS [Ref. 1: p. 117] by doing the matching for him. CAESAR's data will generally be represented as database files. The knowledge base used is a list of decision rules for the assignment process, the majority of which are the dBase II equivalent of IF...THEN statements. CAESAR uses IF statements and data to find a path to the goal state of an optimal officer assignment. It prepares multiple lists of position candidates, based on the degree with which their attributes match the position requirements. The program also divides up the position attributes, assigns values to each, and, by summing them, develops a preference index for each officer selected.

#### B. DESIGN

##### 1. Hardware

The hardware issue requires a detailed cost effectiveness study beyond the scope of this thesis to determine the exact items needed. As an initial cut, however, it

appears that the major database hardware currently used to query the OMF is sufficient. The amount of data maintained on each officer would grow slightly if the DSS is fully implemented, so some additional data storage capacity may be required. Similarly, there should be sufficient hardcopy capacity to give assignment officers file copies of their transactions. Thus some increase in the number of printers in MILPERCEN may be necessary.

## 2. Data

Most of the data for this solution already exists in the OMF and the preference statement of the individual officer. It also includes the position requirements from the major commands mentioned earlier and the current MILPERCEN assignment guidance, some of which will be incorporated into the programs and some of which will be used by the managers to make decisions.

### a. Officer Data Files

TABLE III  
DATABASE RELATIONSHIPS

<u>Name</u>	<u>Origin</u>	<u>Index Keys</u>
ORB	OMF	SSAN, SC1, SC2
ADSPEC		SSAN, SC
PREVSPEC		SSAN, SC
ASI		SSAN
PRC	Major Commands	SC1, SC2, ASI
PREFFORM	DA Form 483	SSAN

There are several files that are needed for this DSS. The most complex is the data shown on the ORB (Figure 2.5). It is basically the extract of the data on each individual that is in the OMF. Much of the data on the ORB is

used for historical purposes or is reviewed for personnel actions other than assignments. In this paper, only those portions relevant to the assignment process will be addressed. These have been placed in dBASE II format for CAESAR's purposes and are linked by the individual's social security account number (SSAN). Their relationships are shown in Table III. These database structures are shown in Appendix C as:

- ADSPEC - Additional Specialties
- ASI - Additional Skill Identifier
- ORB
- PREVSPEC - Previously-held Specialty
- b. PRC File

The next file is the Position Requirement Code (PRC), the exact specifications for the job that the assignment officer is trying to fill. For purposes of this paper, the PRC will be constructed to include all the following data:

- AREA - CONUS or overseas.
- PAN - Command's personnel account number.
- DUTY - type of duty, using the codes from Figure 2.3.
- GRADE - O + a numeral to represent the level of officer required.
- SSI - Specialty Skill Identifier = the basic two digit primary SC1, representing the primary skill required by the job, plus a one letter skill identifier for the subdivision of the SC that would best apply to this position.
- SC2 - Secondary Specialty Code, another SC representing a secondary skill desirable in the nominee. This could be unspecified.
- ASI1 - First Additional Skill Identifier, two characters for a special extra skill required for the position. This could be empty.
- ASI2 - Second ASI, for language or other extra skills that may be required. Also could be blank.
- RPTDATE - Reporting date at this assignment.

An example of a complete PRC and its decryption are found in Appendix C. DUTY is not presently used in

PRC's from the field, but the author proposes it be added to the format to align with preference statement matching and automated career monitoring goals that will be discussed later.

c. PREFFORM File

The "Assignment Preference" section on Figure 2.1 and the questions on Figure 2.2, reveal the data for the file representing the DA Form 483:

- SSAN
- DATE
- PREFSC - Preference for SC assignment.
- PREFSSI - Preference for SSI assignment.
- AREA - CONUS or overseas.
- PRIMACY - Duty or location primary.
- CONUS1 (First preference in CONUS)
- CONUS2
- CONUS3
- LONG1 (First long tour preference)
- LONG2
- SHORT1 (First short tour preference)
- SHORT2
- DUTY1
- DUTY2
- DUTY3
- MILSCHOOL - Desires extra military schooling.
- CIVSCHOOL - Desires postgraduate schooling.
- MAC
- EFM - Exceptional Family Member - special education or medical considerations.
- REMARKS

Codes for location and type duty (Figure 2.3) plus instructions (Figure 2.4) are found on the back of the form.

d. PREFINDX

The final major data element is the preference index. This is the weighted sum computed by CAESAR of all the aspects of an assignment as it relates to the individual's preferences. It is a five-digit number. The higher the number, the more the individual would prefer the assignment.

3. Program

a. Overview

The DSS prototype program is written in dBASE II, since database query is critical to the success of this system and required computations are quite rudimentary.

This program accepts as input the position requirements from the major commands, which are currently sent to MILPERCEN in computer data form. It draws on the OMF for such items as name, SSAN, training, time since last move (to ensure tour equity and stay within minimum tour length guidelines), and school graduation dates. CAESAR matches skills and other attributes to job requirements and then assigns a value to the matching which expresses the nominated officer's relative preference for the assignment.

CAESAR presents the assignment officer with lists of officers who fulfill the job requirements. These lists are ranked by the degree to which the match criteria of Table IV have been met. They also include the preference rating. The lists can be ordered by either preference or last movement date to aid in priority determination. Ideally it will facilitate the assignment of officers to places they have chosen. However, in the event no one has expressed a preference for the position to be filled, CAESAR attempts to optimize the selection of the non-volunteer. For example, if an officer requested a similar assignment in a different country, the preference index points toward him. If a matching is still not possible, then the program

nominates from those available with the required skills, regardless of preference.

TABLE IV  
MATCH CRITERIA

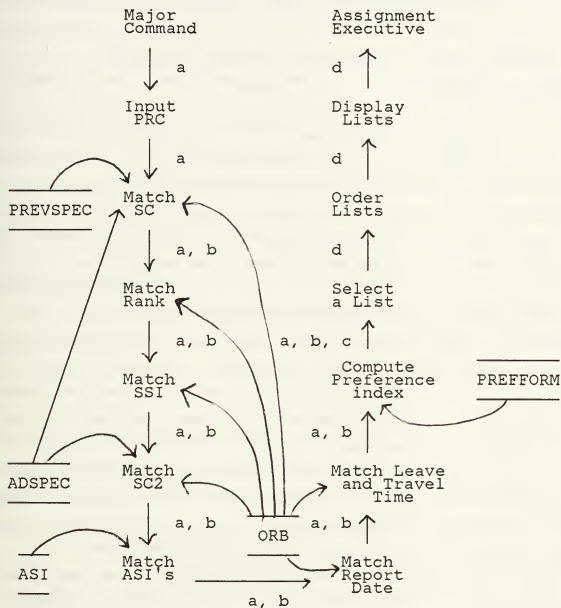
1. Does officer match primary SC?
2. Does officer match primary SC with an old one?
3. Does officer match SSI for required SC ?
4. Does officer match grade (sometimes just within one)?
5. Does officer match secondary SC ?
6. Does officer match primary ASI ?
7. Does officer match secondary ASI ?
8. Does the officer have at least the minimum time between moves ?
9. Does the officer have time for leave and travel between jobs ?

b. Detailed Narrative

The documented source code of CAESAR can be found in Appendix B. An explanatory listing of variables used is located in Appendix C. A narrative explanation of the program's workings follows below. Program flow is depicted in Figure 3.1.

First the user must input the PRC. It can be entered into CAESAR in one of three methods. It can be read in as a database file (DBF), a system data file (SDF), or individual interactive entries. A DBF is a dBASE II database file. A SDF is a regular textfile, in the same general format as the database, that must be run through some dBASE II commands to convert it to a DBF. Interactive input means that the user must fill in each data element as prompted by





#### Data Flows

- a. PRC
- b. Match Information
- c. PREFINDX
- d. Lists

Figure 3.1 Program Flow.

CAESAR. Therefore one of the early screens of the program asks the user to choose his entry method. The DBF/SDF option is used when the PRC data is available to CAESAR in the correct, computerized format. The interactive choice is appropriate when an exceptional request, separate from the normal assignment cycle, comes in and must be processed immediately.

Once the user has chosen the method of inputting the PRC's, CAESAR begins the matching process. The criteria CAESAR uses to screen officers are displayed in Table IV. It looks at one record until it is either rejected or taken all the way through the process and inserted in a list. The primary need is to find an officer of acceptable rank who is qualified in the primary SC of the position. CAESAR queries the OMF, using the SC index, to find the first one which matches the job's primary SC. Then the OMF is searched by officers' secondary SC's to see if any match the primary job SC since officers are considered to be qualified for assignment in either their primary or alternate specialty. Next, if the previous searches have been unsuccessful, any officers with additional specialties that match the position primary SC are queried. Finally, as a last resort, officers listed as having the appropriate SC as a "previously designated specialty" are sought out if there has been no other success. Normally this last category of officer has been classified out of the previously held SC and is no longer considered current and qualified in it. If no officer has been found at this point (almost impossible, given the size of the officer population reflected in the OMF), the job is left vacant until a properly trained officer can be located.

Once an officer has been found, his grade is checked. If it is not the rank called for by the PRC, his name is initially rejected. If no officer of the correct rank can be found, then the program searches for an

appropriately skilled officer one grade junior to the desired grade. The theory here is that a slightly junior officer could learn the job requirements and perhaps be promoted into it later.

If no match can yet be found, the records of officers one rank senior are examined for SC match. If still no match is made, the job is again left temporarily vacant, awaiting the arrival of an appropriately skilled and graded officer. It is felt that an officer two or more grades senior would be severely underemployed in a position and that an officer two or more grades junior to the job requirement would be too inexperienced to be effective in the position. Therefore these officers are not even considered for the post.

Once an officer of some grade has been found qualified in the primary SC of the job, his file in the OMF is further examined to determine how well he fits into the job requirements. While the other requirements of the PRC are not as critical as the primary SC and rank, they are still important in determining who is the best to fill the position. There are nine levels of fit recognized in CAESAR, each with its own list at the end of the process. These categories from top to bottom are shown in Table V.

First the third digit of the SSI is examined to see if the nominee holds that particular skill. Then the job's secondary SC is compared to the primary, alternate, and additional SC's of the officer under consideration. Previously designated SC's are not used here since fineness of fit is being measured so out of date experience is not especially helpful. Next the officer's ASI's in the OMF are compared to the primary and secondary ASI's in the PRC for possible match. These ASI's are normally not key determinants of job qualification because they usually are obtained at a relatively short course of some kind that a nominated

TABLE V  
LIST CHARACTERISTICS

1. Meets all requirements.
2. Meets all requirements except SSI.
3. Meets same requirements as list 2 except for the second ASI.
4. Meets same requirements as list 2 except for the first ASI.
5. Meets same requirements as list 2 except no ASI matches.
6. Meets same requirements as list 5 except no job secondary SC matches.
7. Meets only the SC, grade, and availability requirements.
8. Meets same requirements as list 7 except it uses a previously held SC to meet the SC requirements.
9. Meets only SC and grade requirements.

officer could attend on temporary duty enroute to his new assignment.

Next the officers Date of Estimated Rotation from Overseas (DEROS), or availability date for CONUS-based officers, is evaluated to ensure that the officer will have completed the prescribed minimum length of his previous tour (or graduated from his course of instruction) before having to report for the new job. If no officers were normally available, tours can be curtailed to send an officer to a higher priority assignment. However, in the Gramm-Rudman budget-cutting climate, such additional moves are considered unwise expenditures. Finally the officer's DEROS/availability date is further screened to see if there is sufficient time between assignments for the officer to

take 30 days leave and travel. While this is not a mandatory consideration, it is common to allow time between jobs for vacation and moving. Quick moves, unless at the officer's request, are avoided whenever possible.

When all of these factors have been evaluated there will typically be several officers who qualify, to varying degrees, for the assignment. Now CAESAR takes the officers' personal preferences into account. The preference statement, as mentioned earlier, allows the officer to express a priority between Conus and overseas assignments. It also allows a ranking between duty and location.

Using these choices with the other elements of the Form 483, CAESAR compares the characteristics of the position with the expressed desires of the officer to derive a five-digit preference index. Tables VI and VII show what values CAESAR uses to determine that score.

TABLE VI  
DUTY IS PRIME FACTOR

<u>Match</u>	<u>Value</u>
SC	30000
Secondary SC	15000
Duty 1	3000
2	2000
3	1000
SSI	300
Area	30
CONUS 1	3
2	2
3	1
Overseas 1	3
Short 2	2
Overseas 1	3
Long 2	2
None	0

After the officer has been evaluated as to skill and preference matching, his name is placed on one of the

TABLE VII  
LOCATION IS PRIME FACTOR

<u>Match</u>	<u>Value</u>
Area	30000
CONUS 1	3000
2	2000
3	1000
Overseas 1	3000
Short 2	2000
Overseas 1	3000
Long 2	2000
SC	300
Secondary SC	150
Duty 1	30
2	20
3	10
SSI	3
None	0

nine lists mentioned above, depending on his level of job fit. Then CAESAR examines the next officer, repeating the process until all officers with sufficient matching are on a list. CAESAR next queries the user as to how he wishes the lists to be ordered, by officer preference index or date of last move. The first helps to maximize the value of individual participation, the second aids in checking for tour equity. Once the selection is made, the lists are displayed one at a time on the screen. If a particular level of match is empty, the list is bypassed. All lists with elements are frozen on the screen for examination by the user. Using a "print screen" facility, a hard copy of the list can be acquired, as desired.

Now the user has a listing of all available officers who match the requirements and a concrete indication of their preference for the assignment. This makes the determination of credentials and desires automatic for the assignment officer, simplifying his task. When this

assignment has been taken care of, CAESAR can begin work on the next PRC.

#### 4. Procedures

The individual officer enters his choices via the Form 483. He should update his preferences frequently [Ref. 9: p. 4]. At the receiving end in MILPERCEN, the assignment officer is available to review the individual's desires, if the sending officer requests it, thus assisting the sender in personal career management.

The assignment executive will query CAESAR for nominees for the current positions to be filled. From the output lists, he picks the best qualified officer, based on his current operating guidelines and personal judgment, as the assignment manager does today. The personnel manager should normally start at the top of list 1, since it represents the most highly qualified nominee. If that choice of an officer proves unsatisfactory, the manager goes to the next name on the list. In the event CAESAR delivers a fully qualified list that is empty or the assignment executive does not wish to use any of the officers on it, he is free to march down through the hierarchy of lists until he finds a satisfactory officer. If a personal appeal by an officer in the field for a particular assignment is persuasive to the manager, but CAESAR has not nominated that individual, the assignment officer can also override CAESAR to make a totally manual assignment, as is now the mode. The man controls the machine, but he allows it to make the search effort to find the most qualified nominees. Hopefully, they are volunteers by virtue of their preference statement input. Once an assignment is finalized, the personnel manager updates the database to prevent that officer's name from being used in another assignment. The bulk of the assignment process is unchanged except the computer provides recommendations, biased toward individual skills and

desires, based on a superior ability to keep more variables in "mind" than its human boss.

#### 5. Personnel

Individual officers, field personnel offices, and MILPERCEN workers would require training on the system. No new organizational structures would be required, however. Programmers would require adequate training and documentation to maintain the program.

A minor concern exists about the fairness of this system. Like most systems, CAESAR could be manipulated to reward friends and penalize others. However, that is also certainly true of the manual system. Both the current and the proposed systems depend on the presumed integrity of assignment executives for their smooth execution. Officer integrity is the foundation of the whole military system, however, and must be accepted as a given.

A significant attitude change would be required. MILPERCEN representatives are proud of the fact that Army officers have not been handled by machine, but rather are given the personal touch. Individuals frequently express fear that their lives are being subordinated to computers. However the complexity of the process indicates that the road to optimization is through automation, supervised by caring assignment professionals. Officers, both in the field and at MILPERCEN, would have to be educated along these lines.



#### IV. CONCLUSION AND RECOMMENDATIONS

##### A. CONCLUSION

The Army officer assignment system, while generally functional, is not truly satisfactory, especially with regard to consideration of officer desires and skills. It is feasible to achieve significant improvement through a DSS like CAESAR, supervised by knowledgeable, involved officers. Employment of such a system would greatly improve morale and assignment efficiency plus lower some personnel and training costs.

##### B. RECOMMENDATIONS

###### 1. Implementation

A full-scale DSS to aid the assignment process should be implemented. The production program must be written, as well as accompanying documentation. However, the existence of the CAESAR prototype should ease this process considerably. Much of the hardware, most of the data, the database and network software, the basic assignment and data security procedures, and the operations and user personnel are already in place. A cost/benefit analysis must be done to prove the intuitively appealing contention that the anticipated reduction in personnel and training costs will offset any modest investment required to implement the DSS. The software system should receive some initial testing to avoid immediate alienation of the users. The recommended installation mode is to run the CAESAR and current systems in parallel since, throughout the process, the Army must continue to have its officer requirements met. Since the new system is only a computer-enhanced version of the current process, simultaneous testing and parallel implementation should not be difficult. This plan would

also hasten full operational status for the improved assignment system.

## 2. Preference Sheet Revision

The DA Form 483 should be revised to include all the assignment preference information available on the 1975 edition (Figure 2.6). The DSS could easily be designed to accept the old form's features of 18 choices of duty and location, the additional prioritization between short and long tours, and the separate determination of the primacy of duty or location on each type tour. The availability of all this data would require the designer to make fewer assumptions in the program about the relative importance of these items in computing the preference index, since the submitting officer would be able to more fully present his own ranking of assignments. Thus program results would more accurately represent the desires of the individual officer and increase the probability of his getting the exact assignment he wishes. To achieve these benefits, the only significant costs would be in fielding a revised form, which is a routine operation, and the purchase of any additional storage hardware required to hold the few more spaces per preference record in the OMF database for the additional one- and two-character codes.

## 3. Officer Desires

The role of officer desires should be elevated in the personnel management philosophy, the assignment process, and Army directives. It should be at a level immediately subordinate to Army requirements, above such items as professional development and promotion potential. The needs of the Army are best served by officers who are motivated in their jobs. This is most likely to happen if they choose those jobs for themselves. History has shown that personnel managers have cloudy crystal balls when it comes to predicting future directions for the officer corps and the

tendencies of promotion boards. Since the officer must pay the price of mistakes himself, let him choose what assignments he thinks are "good" for him, if those requirements exist at the appropriate time. If all jobs truly are worth doing, why should an officer be denied one for which he is qualified and must be filled? Commanders, branch and functional area personnel managers, and service school instructors can fulfill their roles in developing the officer corps by advising junior officers of the "correct" career pattern. Professional publications should continue to carry this information and should be widely available. If the individual does not care to avail himself of these resources, he acts at his own risk. Let the promotion process weed out those who stay uninformed or always take the easy jobs. Officers are given full responsibility for the lives of their men and millions of dollars of resources. Why can they not also be fully responsible for their own careers?

#### 4. CAESAR Enhancements

Once the concept of computer-assisted assignments is accepted and the decision has been made to begin design, certain features should be added to the basic CAESAR design.

##### a. List Curtailment

It is possible that the lists requiring the fewest qualifications or the lowest levels of matching could occasionally be hundreds of names long. If the terminal capacity is not large enough for these lists, or it is considered too distracting for the assignment officer, then a routine could be added with a list ceiling of, for example, 20 names. The officers that make the abbreviated lists would be those who would have been the first 20 on their list after the sorting by preference index or date of last move.

#### b. Concurrency

The fully implemented system must provide a mechanism to deal with the problem that several managers could be simultaneously looking at the same group of officers to fill different jobs. Once an officer is given a final assignment, the OMF is updated, but during the nomination process the officer's record can be accessed. An obvious example of this situation would be a branch assignment officer trying to give the individual a position in his primary SC and a functional manager nominating him for a job in his secondary. The system should alert the user to officer names that are being considered in other transactions. Locking the database should be avoided, since many more names will be nominated than used and locks would inhibit multiple concurrent use of the system.

#### c. Measures of Effectiveness

To aid in quantifying the utility of the DSS, a module should be added to compute a degree of preference satisfaction in assignments. A sample metric might be average preference index or how many assignments matched one of the selected officer's preferences. Another computation module should determine how well the program filled the job requirements, such as determining the average matching level of qualification for officers selected for assignments over a given period.

#### d. Career Monitoring

If MILPERCEN is to continue to actively decide the career patterns of officers, modules should be prepared to assist in this effort. The previous assignments of officers (Section IX of Figure 2.5) in the OMF could be coded with duty codes like those used on the DA Form 483 (Figure 2.3). An automated basic screen of an officer's career-to-date could be accomplished using those codes, the Military Education Level (MEL - Section VI of Figure 2.5), and

aviation and other personnel data found in the OMF. A separate routine would have to be prepared for each grade within each branch, since many segments would have different career milestones. This feature should remind the assignment officer, with a remark like "Needs Command," of certain career checkpoints the nominees might be approaching, such as advanced course attendance or flight service "gates," to assist in aligning the next assignment with the currently accepted "correct" career pattern. Other assignment factors such as membership in the MAC or EFM programs could be noted similarly. These routines should have menu-driven maintenance functions to change decision parameters, such as career patterns, since these are subject to routine modification as guidance and Army requirements change. Security measures must be incorporated to ensure these changes are made by authorized personnel only.

e. Regimental Considerations

As the Army converts to the regimental system [Ref. 20], PRC's must indicate the regiment involved, OMF records must also be coded with regimental affiliation, and the DSS must be designed to match an officer's regimental code with that of the PRC to create a new level of fit.

5. Other Assignment Systems

As the DSS shows its value, it should also be applied to the warrant officer assignment system, since it so closely parallels that for commissioned officers. Studies should be done to determine if it can be applied to the non-commissioned officer, junior enlisted, and sister service transfer systems, since they could also benefit by a matching of skills and desires to requirements.

C. OPPORTUNITY

An apparently inexpensive opportunity exists here to use the machine to elevate the role of man in determining his own destiny. Officers will be able to have a more active role in the assignment process than simply shaking their

heads in frustrated disbelief. With due apologies to Senator Edward M. Kennedy, some have looked at the assignment process and said, "Why?" CAESAR examined the system and said, "Why Not?"

## APPENDIX A

### GLOSSARY

ASI	- Additional Skill Identifier
CAESAR	- Commissioned Assignments Executive Support for the US Army
CONUS	- Continental United States
DA	- Department of the Army
DBF	- Database File
DSS	- Decision Support System
EFM	- Exceptional Family Member
MAC	- Married Army Couples
MEL	- Military Education Level
MILPERCEN	- US Army Military Personnel Center
MTOE	- Modification Table of Organization and Equipment
ODP	- Officer Distribution Plan
OMF	- Officer Master File
OPMS	- Officer Personnel Management System
ORB	- Officer Record Brief
PAN	- Personnel Account Number
PRC	- Position Requirements Code
SC	- Specialty Code
SF	- Special Forces
SDF	- System Data File
SSAN	- Social Security Account Number
SSI	- Specialty Skill Identifier
TAADS	- The Army Authorization Document System
TDA	- Table of Distribution and Allowances
TOE	- Table of Organization and Equipment

APPENDIX B  
PROGRAM LISTING

CAESAR uses dBASE II as its language. The program is made up of 16 modules. They are internally documented, though the comments assume the reader has a working knowledge of dBASE II.

The titles, in order of potential execution, are:

```
main.prg
error.prg
inpmenu.prg
prcread.prg
menuread.prg
match.prg
makelist.prg
initial.prg
refine.prg
getback.prg
lists.prg
evaluate.prg
entry.prg
outmenu.prg
display.prg
quit.prg
```



\*\*\*\*\* main.prg \*\*\*\*\*

\* Author: Paul A. Stipek  
 \* Date Written: December 1985  
 \* Purpose: This is the main program for the CAESAR  
 \* system. It associates drives with external  
 \* files and sends the user to a menu.  
 \*

\*\*\*\*\*

```

SET TALK OFF
set deleted on
set escape off
ERASE
store 'Y' to answer
@@ 5,21 SAY "Welcome to CAESAR,"
@@ 7,5 SAY "Commissioned Assignments,Executive "
@@ 7,40 SAY "Support for the US Army."
@@ 10,5 SAY "This system aids US Army Military Personnel "
@@ 10,44 SAY "Center assignment"
@@ 11,5 say "officers match the most qualified.commissioned"
@@ 11,46 SAY "officers"
@@ 12,5 say "for worldwide position requirements. It also "
@@ 12,46 SAY "provides a"
@@ 13,5 say "mechanism for enabling assignment personnel to"
@@ 13,46 SAY "comply with"
@@ 14,5 say "the expressed assignment preferences of the "
@@ 13,44 SAY "officer corps"
@@ 15,5 say "to the maximum extent possible."
@@ 18,12 say "Are you using a color monitor (Y/N) ' get ;
      answer
read
if !(answer) = 'Y'
* Set the character color to bright yellow.
  store 14 to ccolor
* Set the message color to bright yellow on a blue
* background.
  store 30 to mscolor
* Set the error color to flashing red.
  store 140 to errcolor
else
* Set the color to white on black.
  store 7 to ccolor
  store 7 to mscolor
  store 7 to errcolor
endif

set color to 112,ccolor
erase
@@ 10,5 SAY "You will be asked a series of such questions "
@@ 10,45 SAY "by CAESAR."
@@ 11,5 say "A default answer will sometimes be provided "
@@ 11,44 SAY "in the gray"
@@ 12,5 say "entry area. If you agree with that answer, "
@@ 12,44 SAY "just hit enter"
@@ 13,5 say "to respond. If your answer is missing, type "
@@ 13,45 SAY "it in. When"
@@ 14,5 say "you have filled the space, it will "
@@ 14,35 SAY "automatically be entered."
@@ 16,5 SAY "If any of your input is smaller than the size "
@@ 16,46 SAY "of the grey"
@@ 17,5 SAY "entry space provided, type in the characters "
@@ 17,45 SAY "that you need"
@@ 18,5 SAY "to and then hit the return key to move to the "
@@ 18,46 SAY "next prompt."
set color to 112,mscolor
@ 20,5 SAY "Be careful to enter the correct drive "

```

```

@ 20,38 SAY "identifier when asked."
@ 21,5 SAY "An error will terminate the program "
@ 21,36 SAY "immediately with a yellow"
@ 22,5 SAY "dot. Should this happen to you, enter quit "
@ 22,46 SAY "and start the"
@ 23,5 SAY "program again."
set color to ll2, ccolor
store to db:drv
* Ensure correct input.
do while .not. (!(db:drv) = 'A'.or. !(db:drv) = 'B'.or. ;
!(db:drv) = 'C'.or. !(db:drv) = 'D')
@ 6,18 SAY "Which drive has the database ?"
@ 6,51 GET db:drv
set color to ll2, mscolor
@ 8,22 SAY "(Enter A, B, C, or D)"
set color to ll2, ccolor
read
if .not. (!(db:drv) = 'A'.or. !(db:drv) = 'B'.or. ;
!(db:drv) = 'C'.or. !(db:drv) = 'D')
do error
store to db:drv
endif
ENDDO

store db:drv + ':' to db:drv
* Attaching drive information to external files.
set default to &db:drv
* Input preference form data.
store db:drv + 'preform' to preform
* Preform indexed by social security account number (ssan).
store db:drv + 'ssanpref' to ssanpref
* Input Officer Record Brief (orb) personnel data.
store db:drv + 'orb' to orb
* ORB indexed by primary specialty code (sc), an officer's
* main job.
store db:drv + 'scl' to scl
* ORB indexed by secondary sc, an officer's alternate job.
store db:drv + 'sc2' to sc2
* ORB indexed by ssan.
store db:drv + 'ssanorb' to ssanorb
* Input of SC's previously held by the officer.
store db:drv + 'prevspec' to prevspec
* Index by SC for prevspec.
store db:drv + 'scprev' to scprev
* SC's now held by an officer in addition to scl and sc2.
store db:drv + 'adspec' to adspec
* Index by SC for adspec.
store db:drv + 'scad' to scad
* Index by ssan for adspec.
store db:drv + 'ssanad' to ssanad
* Input of additional skill indicators (asi) - special
* training beyond SC's.
store db:drv + 'asi' to asi
* Index by ssan for ASI's.
store db:drv + 'ssanasi' to ssanasi
* An input database of position requirements codes (prc) -
* job descriptors.
* Also serves as a structure repository for use with reqfile
* (below).
store db:drv + 'prc' to prc
* A temporary scratch database to hold input PRC's for
* processing.
store db:drv + 'reqfile' to reqfile
* Serves as a structure repository for use with temporary
* lists which are generated as officer's are matched
* to PRC's.
store db:drv + 'list' to list

use &prc
copy structure to &reqfile

```

```
use
do inpmenu
do quit
```

\*\*\*\*\* error.prg \*\*\*\*\*

\*

Author: Norman Lyons

\*

Date Written: February 1985

\*

Purpose: The error routine flashes a bad input  
message at the corner of the screen and  
beeps three times to let the user know that  
the last command was bad.

\*

\*\*\*\*\*

set color to 112,errcolor

@ 22,64 SAY "Bad Input"

@ 23,64 SAY "Try Again"

@ 23,64 SAY chr(7)

@ 23,64 SAY chr(7)

@ 23,64 SAY chr(7)

set color to 112,mscolor

return

\*\*\*\*\* inpmenu.prg \*\*\*\*\*

```

*
* Author: Paul A. Stipek
*
* Date Written: December 1985
*
* Purpose: This is a menu routine to give the user
*           three choices:
*
*           1. Enter requirements by PRC input file;
*
*           2. Enter requirements interactively;
*
*           3. Quit the program.
*
*

```

\*\*\*\*\*

```

store ' ' to choice
* Loop to allow user to stay in the system for more than
* one choice.
do while !(choice) <> 'Q'
  ERASE
  store ' ' to choice
  * Insure acceptable input.
  do while .not. !(choice) = 'P'.or. !(choice) = 'I'.or. ;
    !(choice) = 'Q'
    set color to 112, ccolor
    @ 11,18 SAY "Which mode do you wish to use?"
    @ 11,56 GET choice
    @ 14,18 SAY "Position Requirement Code (PRC) file "
    @ 14,37 SAY "input,"
    @ 16,18 SAY "Interactive (manual) input, or"
    @ 18,18 SAY "Quit the program?"
    set color to 112, mscolor
    @ 21,26 SAY "(Enter P, I, or Q)"
    set color to 112, ccolor
    read
    if .not. !(choice) = 'P'.or. !(choice) = 'I'.or. ;
      !(choice) = 'Q'
      do error
      store ' ' to choice
    endif
  ENDDO
  do CASE
    CASE !(choice) = 'P'
      do prcread
    CASE !(choice) = 'I'
      do menuread
  ENDCASE
enddo
* Punch out of while loop if choice is 'Q'. Return to main
* program for quit routine call.
return

```

```

***** prcread.prg *****
*
*   Author:   Paul A. Stipek
*
*   Date Written:   December 1985
*
*   Purpose:   This routine is used to read in the position
*               requirements to be filled, from either an
*               SDF file or one in DBF format.
*
*****
set talk off
erase
store 'y' to answer
@ 6,23 SAY "READ IN PRC'S FROM A FILE"
@ 9,5 SAY "This routine reads in position requirements "
@ 9,44 SAY "codes from"
@ 11,5 SAY "a user-supplied file of PRC's for bulk "
@ 11,39 SAY "assignment matching."
@ 13,5 SAY "PRC's in file must be in correct form as per "
@ 13,50 SAY "current directives."
set color to 112,mscolor
@ 22,15 SAY chr(7) + "Do you want to continue (Y/N)?"
@ 22,46 GET answer
set color to 112,ccolor
read
if !(answer) <> 'Y'
    return
endif
erase
store ' ' to db:drv
do while .not. (!(db:drv) = 'A'.or.!(db:drv) = 'B'.or. ;
    !(db:drv) = 'C'.or.!(db:drv) = 'D')
    @ 11,18 SAY "Which drive has your PRC file?"
    @ 11,51 GET db:drv
    set color to 112,mscolor
    @ 13,22 SAY "(Enter A, B, C, or D)"
    set color to 112,ccolor
    read
    if .not. (!(db:drv) = 'A'.or.!(db:drv) = 'B'.or. ;
        !(db:drv) = 'C'.or.!(db:drv) = 'D')
        do error
        store ' ' to db:drv
    endif
endif
ENDDO
store db:drv + ':' to db:drv
set default to &db:drv
erase
store ' ' to sdf
store 'txt' to ext
set color to 112,mscolor
@ 9,13 SAY "Input File Name (up to 8 characters):"
@ 9,49 GET sdf
@ 12,28 SAY "Input File Extension:"
@ 12,49 GET ext
set color to 112,ccolor
@ 20,10 SAY "If any of your input is smaller than the "
@ 20,41 SAY "size of the grey"
@ 21,10 SAY "entry space provided, type in what characters"
@ 21,45 SAY "you need"
@ 22,10 SAY "to and hit the enter key to move to the next "
@ 22,45 SAY "prompt."
set color to 112,ccolor

read
* If DBF file.
if !(ext) = !('dbf')
    store trim(sdf) to sdf
    use &reqfile

```

```

        append from &sdf
    else
        * If SDF file.
        store trim(trim(sdf) + '.' + ext) to sdf
        use &regfile
        append from &sdf sdf
    endif
    * Progress to assignment matching.
    do match
    return

```

\*\*\*\*\* menuread.prg \*\*\*\*\*

```

*
*   Author:   Paul A. Stipek
*
*   Date Written:   December 1985
*
*   Purpose:   This routine is used to input the individual
*               data elements of a position requirements
*               code through an interactive screen filled in
*               by the user. This permits ad hoc searches
*               for job matches for short notice
*               requirements.
*
*****

```

```

set talk off
erase
store 'Y' to answer
@ 6,24 SAY "INTERACTIVE PRC INPUT"
@ 10,7 SAY "This routine permits the user to interactively "
@ 10,47 SAY "query the "
@ 12,7 SAY "Officer Master File to find matches between "
@ 12,44 SAY "officers and "
@ 14,7 SAY "the position requirements entered by the user. "
@ 14,47 SAY "One PRC"
@ 16,7 SAY "can be processed at a time. If you wish to "
@ 16,44 SAY "process "
@ 18,7 SAY "additional requirements, you will be given an "
@ 18,46 SAY "opportunity "
@ 20,7 SAY "after each requirement is processed."
set color to 112,mscolor
@ 23,15 SAY chr(7) + "Do you want to continue (Y/N)?"
@ 23,45 GET answer
set color to 112,ccolor
read
if !(answer) <> 'Y'
    return
endif
erase
use &regfile
append blank

```

```

* Default values to help in data entry error reduction.
replace area with '0'
replace pan with '00'
replace duty with '0'
replace grade with '00'
replace scl with '00'
replace ssi with '0'
replace sc2 with '00'
replace asi1 with '00'
replace asi2 with '00'

```

```

@ 1,17 SAY "PRC Entry Screen"
@ 4,10 SAY "Area"
@ 4,43 GET area
@ 6,10 SAY "Personnel account number"
@ 6,43 GET pan
@ 8,10 SAY "Type of duty to be filled"
@ 8,43 GET duty
@ 10,10 SAY "Grade required"
@ 10,43 GET grade
@ 12,10 SAY "Primary specialty code"
@ 12,43 GET scl
* SSI is a subcategory of SC - generally not very
* significant in the process.
@ 14,10 SAY "Special skill identifier"
@ 14,43 GET ssi

```



```

@ 16,10 SAY "Secondary specialty code"
@@ 16,43 GET sc2
@@ 18,10 SAY "First additional skill"
@@ 18,43 GET asil
@ 20,10 SAY "Second additional skill"
@ 20,43 GET asi2

set color to ll2, mscolor
@@ 21, 10 SAY "If any of your input is smaller than the size"
@@ 21, 45 SAY " of the grey"
@@ 22, 10 SAY "entry space provided, after you have typed "
@@ 22, 43 SAY "what you need to"
@@ 23, 10 SAY "hit the enter key to move to the next data "
@ 23, 43 SAY "prompt. "
set color to ll2, ccolor
read
* Progress to matching of PRC's and assignments.
do match
return

```

\*\*\*\*\* match.prg \*\*\*\*\*

```

*
*   Author:   Paul A. Stipek
*
*   Date Written:  December 1985
*
*   Purpose:  This routine performs the gross officer to
*             job requirement matching.  It looks at
*             specialty codes (SC) and special skill
*             indicators (SSI) plus arranges looping as
*             required for grade/rank matching.  It calls
*             other routines to refine the selection.
*
*****

```

```

erase
set color to 112, mscolor
@ 10, 5 SAY "Please be patient.  CAESAR is matching the "
@ 10, 43 SAY "position requirement"
@ 12, 5 SAY "code (PRC) with officer skills and desires to "
@ 12, 46 SAY "produce"
@ 14, 5 SAY "the best matches, so this may take a few "
@ 14, 41 SAY "minutes."
set color to 112, ccolor
store 0 to count
store f to finished
* If correct grade cannot be matched, one down and one up
* can be used.
store f to junior
store f to senior
* This is the file with the PRC to be filled.
select primary
use &regfile
* Loop for multiple prc's in the file.
do while .not. eof
    store # to regnum
    * Create a set of nine lists representing levels of officer
    * matching to the PRC.
    do makelist

    * Loop if rank must be varied.
    do while .not. finished
        select secondary
    * Officer data to try to match the prc's with.
        use 2.5 index &sc1, &sc2, &ssanorb
        store p.sc1 to key
        find &key
    * If no one matches the primary sc, set the flag to keep
    * looking.
        if # = 0
            store t to need:one
    * Got one.
        else
            store f to need:one
    * Loop through all officers that might match.
        do while !(p.sc1) = !(s.sc1) .and. .not. eof
    * Initializes boolean flags for go/no-go matching.
            do initial
    * Officer has the right primary SC.
            store t to ok:sc
            if !(ssi) = !(ssil)
    * SSI match; SSI's are paired with specific SC's since
    * they are subdivisions.
    * To match ssil with sc2 would be meaningless.
                store t to ok:ssi
            endif
    * Already have at least one at this point, now see how well
    * he fits.

```

```

do refine
* Refine.prg will increment count to keep track of how many
* matches occur.
    skip
enddo
    if count = 0
* Set flag to keep looking.
    store t to need:one
    endif
endif

select secondary
* Similar to process above except look at sc2 here.
* Current policy is to treat an officer as fully qualified
* in both sc1 and sc2.
    use 2.5 index &sc2, &sc1, &ssanorb
    store p.sc1 to key
    find &key
    if # <> 0
        store f to need:one
        do while !(p.sc1) = !(s.sc2) .and. .not. eof
            do initial
                store t to ok:sc
                if !(ssi) = !(ssi2)
                    store t to ok:ssi
                endif
            do refine
                skip
            enddo
            if count = 0
                store t to need:one
            endif
        endif
* If still no match, look at additional specialties.
        if need:one
            select secondary
            use &adspec index &scad, &ssanad
            store p.sc1 to key
            find &key
            if # <> 0
                store f to need:one
                do while !(p.sc1) = !(sc) .and. .not. eof
                    do initial
                        store t to ok:sc
                        if !(p.ssi) = !(s.ssi)
                            store t to ok:ssi
                        endif
                    do refine
                        store s.ssan to key2
                    select secondary
                    use 2.5 index &ssanorb, &sc1, &sc2
                    find &key2
                    do refine
                        skip
                    enddo
                    if count = 0
                        store t to need:one
                    endif
                endif
            endif
        endif
* Need a third work area for the ORB, so capture the key
* before leaving.
        store s.ssan to key2
        select secondary
        use 2.5 index &ssanorb, &sc1, &sc2
        find &key2
        do refine
            skip
        enddo
        if count = 0
            store t to need:one
        endif
    endif
endif

* If still no luck, look at previous specialties, if any.
* Last resort because officer will probably not be current
* in this sc.
    if need:one
        select secondary
        use &prevspec index &scprev
        store p.sc1 to key
        find &key
    endif

```

```

    if # <> 0
        store f to need:one
        do while !(p.scl) = !(sc) .and. .not. eof
            do initial
                store t to ok:sc
                if !(p.ssi) = !(s.ssi)
                    store t to ok:ssi
                endif
                store s.ssan to key2
                select secondary
                use 2.5 index &ssanorb, &scl, &sc2
                find &key2
                do refine
                    skip
                enddo
                if count = 0
                    store t to need:one
                endif
            endif
        endif
    * Represents unsuccessful look at plus and minus one rank
    * as well.
        if (need:one .and. senior)
            store t to finished
        else
    * Represents unsuccessful look at minus one rank.
            if (need:one .and. junior)
                store t to senior
                store f to junior
            else
    * Represents unsuccessful look at requested rank.
                if need:one
                    store t to junior
                endif
            endif
        endif
    * If successful.
        if .not. need:one
            store t to finished
        endif
    enddo
    * Can't fill it today.
        if (finished .and. need:one)
            erase
            set color to 112, errcolor
            @ 5, 10 SAY "No qualified officers available at this "
            @ 5, 40 SAY "time"
            @ 6, 10 SAY "for PRC "
            @ 6, 18 SAY regnum + " "
            set color to 112, mscolor
        endif
    * Type out the ordered lists of nominees.
        do outmenu
        do getback
        skip
    * Move on to next PRC in reqfile, if have another.
    enddo
    return

```

```

***** makelist.prg *****
*
*   Author:   Paul A. Stipek
*
*   Date Written:   December 1985
*
*   Purpose:   This routine performs the initial structuring
*               of the nine selection lists required when a
*               new PRC is under consideration.
*
*****

* Lists are developed in lists.prg.
* list1 represents the highest level of matching; list9 the
* least.

store db:drv + 'list1' to list1
store db:drv + 'list2' to list2
store db:drv + 'list3' to list3
store db:drv + 'list4' to list4
store db:drv + 'list5' to list5
store db:drv + 'list6' to list6
store db:drv + 'list7' to list7
store db:drv + 'list8' to list8
store db:drv + 'list9' to list9
store 0 to counter
do while counter <= 9
  store counter + 1 to counter
  do case
    case counter = 1
      store list1 to listname
    case counter = 2
      store list2 to listname
    case counter = 3
      store list3 to listname
    case counter = 4
      store list4 to listname
    case counter = 5
      store list5 to listname
    case counter = 6
      store list6 to listname
    case counter = 7
      store list7 to listname
    case counter = 8
      store list8 to listname
    case counter = 9
      store list9 to listname
  endcase
  select secondary
  use &list
  copy structure to &listname
  use
enddo
return

```

```

***** initial.prg *****
*
*   Author:   Paul A. Stipek
*
*   Date Written:   December 1985
*
*   Purpose:   This routine initializes the boolean flags
*               before each officer's file is checked for
*               matching.
*
*****

* Following comments explain the boolean variables.
* Does officer match primary SC?
store f to ok:sc
* Does officer match primary SC with an old one?
store f to ok:prev
* Does officer match grade (some times just within one)?
store f to ok:grade
* Does officer match SSI for required SC ?
store f to ok:ssi
* Does officer match secondary SC ?
store f to ok:sc2
* Does officer match primary ASI ?
store f to ok:asil
* Does officer match secondary ASI ?
store f to ok:asi2
* Does the officer have at least the minimum time between
* moves ?
store f to ok:min
* Does the officer have time for leave and travel between
* jobs ?
store f to ok:lvttl

return

```

```

***** refine.prg *****
*
*   Author:   Paul A. Stipek
*
*   Date Written:   December 1985
*
*   Purpose:   This routine completes the detailed
*               comparison of the officer's characteristics
*               with the position requirements and calls
*               some the output routines.
*
*****

* Primary here is still reqfile; secondary is ORB,
* indexed by one of three fields, ssan, sc1, sc2.
* Plus one rank here okay.
if senior
  if val($(p.grade,2,1)) + 1 = val($(s.grade,2,1))
    store t to ok:grade
  else
    return
  endif
else
  * Minus one rank here okay.
  if junior
    if val($(p.grade,2,1)) - 1 = val($(s.grade,2,1))
      store t to ok:grade
    else
      return
    endif
  else
    * Correct grade here.
    if !(p.grade) = !(s.grade)
      store t to ok:grade
    else
      * No luck here; no more variation than one rank; no
      * colonels to 2LT jobs or vice versa.
      return
    endif
  endif
endif

* sc2 not important to this job PRC.
if p.sc2 = "00"
  store t to ok:sc2
else
  * If job's secondary matches officer's primary.
  if !(p.sc2) = !(s.sc1)
    store t to ok:sc2
  else
    if !(p.sc2) = !(s.sc2)
      store t to ok:sc2
    else
      * Hold job SC2 here while switching primary database.
      store p.sc2 to temp
      select primary
      * Try adspecs; will not look at prevspec because of
      * currency problem.
      * Don't get to here until at least one match so don't need
      * noncurrent officers.
      use &adspec index &ssanad, &scad
      store s.ssana to key
      find &key
      if # <> 0
        do while p.ssana = s.ssana .and. .not. eof
          if !(temp) = !(p.sc)
            store t to ok:sc2
          endif
          skip
        endwhile
      endif
    endif
  endif
endif

```

```

        enddo
      endif
* Recover the old primary work area.
      do getback
        release temp
      endif
    endif
  endif

* Now the same drill for ASI matches.
if p.asi1 = '00'
  store t to ok:asi1
else
* ASI data resides on a separate database so switch again.
  store p.asi1 to temp
  select primary
  use &asi index &ssanasi
  store s.ssan to key
  find &key
  if # <> 0
    do while .not. eof .and. p.ssan = s.ssan
      if !((temp) = !(asi))
        store t to ok:asi1
      endif
      skip
    enddo
  endif
  do getback
  release temp
endif

if p.asi2 = '00'
  store t to ok:asi2
else
  store p.asi2 to temp
  select primary
  use &asi index &ssanasi
  store s.ssan to key
  find &key
  if # <> 0
    do while .not. eof .and. p.ssan = s.ssan
      if !((temp) = !(asi))
        store t to ok:asi2
      endif
      skip
    enddo
  endif
  do getback
  release temp
endif

* Date of Estimated Return from OverSeas (deros).
* Checking to see if officer overseas will be finished in
* time to take this assignment.

if (deros > 0 .and. deros <= rptdate)
  store t to ok:min
endif

* Availability date for stateside officers - their release
* date as determined by graduation dates, stabilization
* requirements, and minimum time-on-station policies - cost
* control measures.

if (availdate > 0 .and. availdate <= rptdate)
  store t to ok:min
endif

* Now look to squeeze 30 days leave in.
* Because of the yymmdd format, 100 represents one month.

```



```

* A January rptdate would lead to a 00 month; while
* artificial, this does not upset program logic.

if (deros > 0 .and. deros <= rptdate - 100)
  store t to ok:lvttl
endif

if (availdate > 0 .and. availdate <= rptdate - 100)
  store t to ok:lvttl
endif

* Update the count of officer matches (to varying degrees).
store count + 1 to count
* Plug the officer into a list based on those varying
* degrees.
do lists
return

```

```

***** getback.prg *****
*
*   Author:  Paul A. Stipek
*
*   Date Written:  December 1985
*
*   Purpose:  This routine returns regfile to position as
*             primary database after temporary
*             displacement and positions back to the PRC
*             under consideration before the interrupt.
*
*****

select primary
use &regfile
goto regnum
return

```

```

***** lists.prg *****
*
*   Author:   Paul A. Stipek
*
*   Date Written:  December 1985
*
*   Purpose:   This routine determines the appropriate list
*               for a qualified officer. List1 equates to
*               maximum matching of the PRC by the officer;
*               list9 represents a minimal match.
*
*****

* Determine how well the assignment matches the nominated
* officer's preferences.
do evaluate

* If make one list, the officer is screened from
* all others.

* Matches all requirements; SSI represents SC + SSI pair.
if ok:ssi .and. ok:grade .and. ok:sc2 .and. ok:asil .and. ;
   ok:asi2 .and. ok:min .and. ok:lvtvl
   store list1 to listname
   do entry
else
* Match all except SSI.
   if ok:sc .and. ok:grade .and. ok:sc2 .and. ok:asil ;
      .and. ok:asi2 .and. ok:min .and. ok:lvtvl
      store list2 to listname
      do entry
   else
* asi2 drops out.
      if ok:sc .and. ok:grade .and. ok:sc2 .and. ok:asil ;
         .and. ok:min .and. ok:lvtvl
         store list3 to listname
         do entry
      else
* asil drops out.
         if ok:sc .and. ok:grade .and. ok:sc2 .and. ;
            ok:asi2 .and. ok:min .and. ok:lvtvl
            store list4 to listname
            do entry
         else
* Both asi's gone.
            if ok:sc .and. ok:grade .and. ok:sc2 .and. ;
               ok:min .and. ok:lvtvl
               store list5 to listname
               do entry
            else
* sc2 gone.
               if ok:sc .and. ok:grade .and. ok:min .and. ;
                  ok:lvtvl
                  store list6 to listname
                  do entry
               else
* No time for leave/travel.
                  if ok:sc .and. ok:grade .and. ok:min
                     store list7 to listname
                     do entry
                  else
* Uses previous SC to match.
                     if ok:prev .and. ok:grade .and. ok:min
                        store list8 to listname
                        do entry
                     else
* Only an acceptable grade and SC.
                        if ok:sc .and. ok:grade

```

```
store list9 to listname
```

endif

```
endif
```

endif

endif

endif

endif

```
endif
```

```
endif
```

```

enc
endif

```

```
endif
return
```

\*\*\*\*\* evaluate.prg \*\*\*\*\*

\* Author: Paul A. Stipek  
\* Date Written: December 1985  
\* Purpose: This routine determines the individuals  
\* preference index for the assignment under  
\* consideration by comparing his stated  
\* preferences for duty and location with the  
\* characteristics of the PRC. A five-digit  
\* number is used to quantify a relative  
\* preference.

\*\*\*\*\*

```
do getback
store s.ssan to temp
select secondary
use &prefform index &ssanpref
find &temp
store 0 to rating
* If duty is a greater consideration to the officer than
* location, the high order digits will be based on job
* characteristics and not location matching.

if !(primacy) = 'D'
  if !(scl) = !(prefsc)
    store rating + 30000 to rating
  else
    if !(sc2) = !(prefsc)
      store rating + 15000 to rating
    endif
  endif
endif

if !(duty) = !(duty1)
  store rating + 3000 to rating
else
  if !(duty) = !(duty2)
    store rating + 2000 to rating
  else
    if !(duty) = !(duty3)
      store rating + 1000 to rating
    endif
  endif
endif

if !(scl + ssi) = !(prefsc + prefssi)
  store rating + 300 to rating
endif

* Now location considerations.
  if !(p.area) = !(s.area)
    store rating + 30 to rating
  endif
* If first choice is stateside (CONUS) duty.
  if !(s.area) = 'C'
    if !(pan) = !(conus1)
      store rating + 3 to rating
    else
      if !(pan) = !(conus2)
        store rating + 2 to rating
      else
        if !(pan) = !(conus3)
          store rating + 1 to rating
        else
          * Long (three year) overseas tours.
          if !(pan) = !(long1)
```

```

        store rating + 3 to rating
    else
        if !(pan) = !(long2)
            store rating + 2 to rating
        else
* Short (one year) overseas tours.
            if !(pan) = !(short1)
                store rating + 3 to rating
            else
                if !(pan) = !(short2)
                    store rating + 2 to rating
                endif
            endif
        endif
    endif
endif
endif
endif
else
* Overseas before CONUS.
    if !(pan) = !(long1)
        store rating + 3 to rating
    else
        if !(pan) = !(long2)
            store rating + 2 to rating
        else
            if !(pan) = !(short1)
                store rating + 3 to rating
            else
                if !(pan) = !(short2)
                    store rating + 2 to rating
                else
                    if !(pan) = !(conus1)
                        store rating + 3 to rating
                    else
                        if !(pan) = !(conus2)
                            store rating + 2 to rating
                        else
                            if !(pan) = !(conus3)
                                store rating + 1 to rating
                            endif
                        endif
                    endif
                endif
            endif
        endif
    endif
endif
endif
else
* In this region, location is a higher preference than duty.
    if !(p.area) = !(s.area)
        store rating + 30000 to rating
    endif
    if !(s.area) = 'C'
        if !(pan) = !(conus1)
            store rating + 3000 to rating
        else
            if !(pan) = !(conus2)
                store rating + 2000 to rating
            else
                if !(pan) = !(conus3)
                    store rating + 1000 to rating
                else
                    if !(pan) = !(long1)
                        store rating + 3000 to rating
                    else
                        if !(pan) = !(long2)
                            store rating + 2000 to rating
                        else

```



```
***** entry.prg *****
*
*   Author:   Paul A. Stipek
*
*   Date Written:  December 1985
*
*   Purpose:   This routine enters a qualified officer's
*               pertinent statistics in a list appropriate
*               to his selection category.
*
*****
```

```
select primary
use &listname
* Add a new person to the list.
append blank
select secondary
use 2.5 index &ssanorb, &sc1, &sc2
* ORB ssan from evaluate.prg held in temp during the
* rating computation.
find &temp
* Fill in the blank record.
select primary
replace p.lastn with s.lastn
replace p.firstn with s.firstn
replace p.ssan with s.ssan
replace p.grade with s.grade
replace p.branch with s.branch
replace p.sc1 with s.sc1
replace p.sc2 with s.sc2
replace p.lastpcs with s.lastpcs
replace prefindx with rating
return
```



\*\*\*\*\* outmenu.prg \*\*\*\*\*

```

*
*   Author:   Paul A. Stipek
*
*   Date Written:  December 1985
*
*   Purpose:   This is a menu routine to give the user five
*              choices of post-matching activity:
*
*              1.  View selected lists in preference
*                  order,
*
*              2.  View selected lists in last move
*                  date order,
*
*              3.  Delete the lists,
*
*              4.  Return to process the next PRC, or
*
*              5.  Quit the program.
*
*
*
*
*
*
*
*
*
*

```

\*\*\*\*\*

```

store ' ' to choice
* Loop to allow user to stay in the system for more than
* one choice.
do while !(choice) <> 'R' .and. !(choice) <> 'Q'
  ERASE
  store ' ' to choice
  do while .not. (!{choice} = 'P'.or. !{choice} = 'L'.or. ;
    !{choice} = 'D'.or. !{choice} = 'R'.or. ;
    !{choice} = 'Q')
    set color to 112, ccolor
    @ 5,18 SAY "Which activity do you wish next?"
    @ 5,56 GET choice
    @ 8,5 SAY "Preference-ordered display of the names "
    @ 8,40 SAY "on each list,"
    @ 10,5 SAY "Last PCS date-ordered display of the "
    @ 10,37 SAY "names on each list,"
    @ 12,5 SAY "Deletion of the lists (this eliminates "
    @ 12,39 SAY "the previous two),"
    @ 14,5 SAY "Return to process the next PRC, or"
    @ 16,5 SAY "Quit the program?"
    set color to 112, mscolor
    @ 20,26 SAY "(Enter P, L, D, R, or Q)"
    set color to 112, ccolor
    read
    if .not. (!{choice} = 'P'.or. !{choice} = 'L'.or. ;
      !{choice} = 'D'.or. !{choice} = 'R'.or. ;
      !{choice} = 'Q')
      do error
      store ' ' to choice
    endif
  ENDDO
  do CASE
    CASE !{choice} = 'P'
      do display
    CASE !{choice} = 'L'
      do display
    CASE !{choice} = 'D'
      do display
    CASE !{choice} = 'R'
      return
    CASE !{choice} = 'Q'
      store t to eof
      return
  ENDCASE
enddo

```

- \* Punch out of while loop if choice is 'R' or 'Q'.
- \* Return to match to continue processing or main program
- \* for quit routine call.

```

***** display.prg *****
*
* Author: Paul A. Stipek
*
* Date Written: December 1985
*
* Purpose: This routine displays the nominee lists in the
* requested indexed order or deletes files.
*
*****
store 0 to counter
* Loop through all lists.
do while counter < 9
  store counter + 1 to counter
  do case
    case counter = 1
      store list1 to listname
    case counter = 2
      store list2 to listname
    case counter = 3
      store list3 to listname
    case counter = 4
      store list4 to listname
    case counter = 5
      store list5 to listname
    case counter = 6
      store list6 to listname
    case counter = 7
      store list7 to listname
    case counter = 8
      store list8 to listname
    case counter = 9
      store list9 to listname
  endcase
  erase
  use &listname
* Delete all lists, one at a time.
  if !(choice) = 'D'
    delete file &listname
  else
* List names in the order they desire the assignment based
* on prefindx.
    if !(choice) = 'P'
      index on prefindx to prefindx
      store db:drv + prefindx to prefindx
      use &listname index &prefindx
    else
* List names in the order based on when they last moved.
      index on lastpcx to pcsindx
      store db:drv + pcsindx to pcsindx
      use &listname index &pcsindx
    endif
  endif
* Print each list out, on the terminal.
  if !(choice) <> 'D'
    erase
    store 0 to line
    if # <> 0
      @ 1, 22 SAY "DEGREE OF PRC MATCH: Level "
      @ 1, 49 SAY $(listname, 7, 1)
      @ 2, 26 SAY "(Level 1 = maximum)"
      @ 4, 8 SAY "Name"
      @ 4, 32 SAY "Grade Br. Pri. 2nd PCS'd" SSAN "
      @ 4, 67 SAY "Score"
    do while .not. eof
      store line + 1 to line
      store trim (lastn) + ' ' + trim (firstn) to ;
        nameline
    
```

```

        @ 6 + line , 1 say nameline
        @ 6 + line , 30 say ssan
        @ 6 + line , 42 say grade
        @ 6 + line , 46 say branch
        @ 6 + line , 51 say scl
        @ 6 + line , 55 say sc2
        @ 6 + line , 59 say lastpcs
        @ 6 + line , 67 say prefindx
        skip
    enddo
    set color to ll2, mscolor
    @ 23, 8 SAY chr (7) + "After the disc stops (red "
    @ 23,26 SAY "light out ),"
    @ 23,46 SAY "press any key to continue."
    set console off
    wait
    set console on
    set color to ll2, ccolor
endif
endif
erase
* Empty lists appear as blank screens; so statement used to
* show that the computer is awake.
    @ 10,10 SAY "CAESAR examines each list to carry out "
    @ 10,44 SAY "your request."
enddo
return

```

```
***** quit.prg *****
*
*   Author:   Paul A. Stipek
*
*   Date Written:  December 1985
*
*   Purpose:   This program terminates processing and
*              returns control to the operating system.
*
*****
```

```
erase
@ 10,21 SAY "End of CAESAR."
set color to 112,mscolor
@ 20,11 SAY chr(7)+"Press any key after the disk stops (red"
@ 20,39 SAY "light out)"
@ 21,16 SAY "to return control to the operating system."
set color to 112,ccolor
set console off
wait
set console on
quit
```

# APPENDIX C DATA DICTIONARY

1. The first items listed are the database structures used in CAESAR and explained in Chapter III:

## Format Definitions

FLD - Field identification number.

NAME - Title of field.

TYPE - Type of data in the field.  
C - Character  
L - Logical  
N - Numeric

WIDTH - Number of positions used by the field.

DEC - Number of decimal places for numeric data.

STRUCTURE FOR FILE:	B:ADSPEC	.DBF	Additional
NUMBER OF RECORDS:	00001		Specialty
PRIMARY USE DATABASE			Codes
FLD	NAME	TYPE WIDTH DEC	
001	SSAN	C 009	
002	SC	C 002	
003	SSI	C 001	
** TOTAL **		00013	

STRUCTURE FOR FILE:	B:ASI	.DBF	Additional
NUMBER OF RECORDS:	00013		Skill
PRIMARY USE DATABASE			Identifiers
FLD	NAME	TYPE WIDTH DEC	
001	SSAN	C 009	
002	ASI	C 002	
** TOTAL **		00012	

STRUCTURE FOR FILE:	B:LIST	.DBF	Structure for
NUMBER OF RECORDS:	00000		the officer
PRIMARY USE DATABASE			nominee lists
FLD	NAME	TYPE WIDTH DEC	to be developed
001	LASTN	C 020	
002	FIRSTN	C 020	
003	SSAN	C 009	
004	GRADE	C 002	
005	BRANCH	C 002	
006	SC1	C 002	
007	SC2	C 002	
008	LASTPCS	N 006	Date last moved
009	PREFINDX	N 005	Preference score
** TOTAL **		00069	

STRUCTURE FOR FILE: B:ORB .DBF Officer Record  
NUMBER OF RECORDS: 00005 Brief  
PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH	DEC
001	SSAN	C	009	
002	LASTN	C	020	
003	FIRSTN	C	020	
004	MIDDLE	C	020	
005	GRADE	C	002	
006	BRANCH	C	002	
007	CONTROL	C	002	
008	LASTUPDATE	N	006	
009	SHORT	N	002	
010	LONG	N	002	
011	DROS	N	006	
012	DEROS	N	006	
013	CLEARANCE	C	002	
014	SEX	C	001	
015	FAMILY	N	002	
016	MARITAL	C	001	
017	PULHES	C	006	
018	ADOR	N	006	
019	SC1	C	002	
020	SSI1	C	001	
021	SC2	C	002	
022	SSI2	C	001	
023	MEL	C	001	
024	CEL	C	001	
025	AVAILDATE	N	006	
026	LASTPCS	N	006	
027	PSC	C	001	
028	ASED	N	006	
029	TOFDC	N	002	
030	FDCDATE	N	006	
031	TFOS	N	006	
** TOTAL **			00157	

STRUCTURE FOR FILE: B:PRC .DBF Position  
NUMBER OF RECORDS: 00001 Requirements  
PRIMARY USE DATABASE Code - job  
descriptions

FLD	NAME	TYPE	WIDTH	DEC
001	AREA	C	001	
002	PAN	C	002	
003	DUTY	C	001	
004	GRADE	C	002	
005	SC1	C	002	
006	SSI	C	001	
007	SC2	C	002	
008	ASI1	C	002	
009	ASI2	C	002	
010	RPTDATE	N	006	
** TOTAL **			00022	

STRUCTURE FOR FILE: B:PREFFORM.DBF  
NUMBER OF RECORDS: 00003  
PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH	DEC
001	SSAN	C	009	
002	DATE	N	006	
003	PREFSC	C	002	
004	PREFSSI	C	001	
005	AREA	C	001	
006	PRIMACY	C	001	
007	CONUS1	C	002	
008	CONUS2	C	002	
009	CONUS3	C	002	
010	LONG1	C	002	
011	LONG2	C	002	
012	SHORT1	C	002	
013	SHORT2	C	002	
014	DUTY1	C	001	
015	DUTY2	C	001	
016	DUTY3	C	001	
017	MILSCHOOL	C	001	
018	CIVSCHOOL	C	001	
019	MAC	C	001	
020	EFM	C	001	
021	REMARKS	C	001	
** TOTAL **			00043	

Officer  
Assignment  
Preference  
Statement -  
DA Form 483

STRUCTURE FOR FILE: B:PREVSPEC.DBF  
NUMBER OF RECORDS: 00001  
PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH	DEC
001	SSAN	C	009	
002	SC	C	002	
003	SSI	C	001	
** TOTAL **			00013	

Previously  
held SC's

STRUCTURE FOR FILE: B:REOFILE.DBF  
NUMBER OF RECORDS: 00001  
PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH	DEC
001	AREA	C	001	
002	PAN	C	002	
003	DUTY	C	001	
004	GRADE	C	002	
005	SC1	C	002	
006	SSI	C	001	
007	SC2	C	002	
008	ASI1	C	002	
009	ASI2	C	002	
010	RPTDATE	N	006	
** TOTAL **			00022	

Structure that  
SDF files like  
test.txt (see  
paragraph 3.)  
are placed in  
for processing



2. Next listing is that of the memory variables used in CAESAR with typical values:

<u>Name</u>	<u>Type</u>	<u>Example</u> <u>Value</u>	<u>Comments</u>
adspec	(c)	b: adspec	complete filename
answer	(c)	y	user response
asi	(c)	b: asi	filename
ccolor	(n)	14	character color
choice	(c)	q	menu choice
count	(n)	1	success total
counter	(n)	9	incrementer
db: drv	(c)	b:	drive prefix
eof	(l)	.t.	end of file test
errcolor	(n)	140	error color
ext	(c)	txt	file type
finished	(l)	.t.	boolean flag
junior	(l)	.f.	boolean flag
key	(c)	15	search variable
line	(n)	0	output incrementer
list	(c)	b: list	filenames
listname	(c)	b: list9	
list1	(c)	b: list1	
list2	(c)	b: list2	
list3	(c)	b: list3	
list4	(c)	b: list4	
list5	(c)	b: list5	
list6	(c)	b: list6	
list7	(c)	b: list7	
list8	(c)	b: list8	
list9	(c)	b: list9	
mscolor	(n)	30	message color
nameline	(c)	stipek, paul	officer
need: one	(l)	.f.	boolean flags

ok: asil	(1)	. t.	
ok: asi2	(1)	. t.	
ok: grade	(1)	. t.	
ok: lvtv1	(1)	. t.	
ok: min	(1)	. t.	
ok: prev	(1)	. f.	
ok: sc	(1)	. t.	
ok: sc2	(1)	. t.	
ok: ssi	(1)	. f.	
orb	(c)	b: orb	filenames
prc	(c)	b: prc	
prefform	(c)	b: prefform	
prefindx	(c)	b: prefindx	
prevspec	(c)	b: prevspec	
rating	(n)	19020	preference index
reqfile	(c)	b: reqfile	filename
regnum	(n)	1	current record
scad	(c)	b: scad	filenames
scprev	(c)	b: scprev	
scl	(c)	b: scl	
sc2	(c)	b: sc2	
sdf	(c)	test.txt	
senior	(1)	. f.	boolean flag
ssanad	(c)	b: ssanad	filenames
ssanasi	(c)	b: ssanasi	
ssanorb	(c)	b: ssanorb	
ssanpref	(c)	b: ssanpref	
temp	(c)	033384357	search variable
** total **      57 variables used			

3. One textfile was used for test data, test.txt.  
It contains a PRC to be filled:

cnfao415b181n5g860430

Applying the PRC format, this translates into:

c        - Stateside (CONUS) area  
nf       - 1st Special Operations Command (SOCOM),  
          Fort Bragg, N.C.  
a        - command duty  
o4       - major  
15b      - combat aviation officer  
18       - special operations officer  
1n       - UH-60 Blackhawk pilot  
5g       - Special Forces (SF) qualified  
860430 - 30 April 1986

Thus SOCOM is looking for an aviator major who is also a special operations type, trained in the UH-60 helicopter and SF, for duty as a unit commander, reporting on 30 April 1986.

APPENDIX D  
SAMPLE OUTPUT

DEGREE OF PRC MATCH: Level 2  
(Level 1 = maximum)

Name	SSAN	Grade	Br.	Pri.	2nd	PCS'd	Score
stipek, paul	033384357	o4	av	15	53	840820	19020

## LIST OF REFERENCES

1. Keen, P. and Wagner, G., "DSS: An Executive Mind-Support System," DATAMATION, November, 1979.
2. Kroenke, D., Business Computer Systems, An Introduction, 2d ed., Mitchell Publishing, Inc., 1984.
3. Department of the Army Regulation 611-101, Commissioned Officer Specialty Classification System, 30 October 1984.
4. "OPMS Study Group," Army Times, 8 October 1984.
5. Tice, J., "How to Get Right Job Next Tour," Army Times, 17 June 1985.
6. Department of the Army Regulation 623-105, Officer Evaluation Reporting System, 15 November 1981.
7. Tice, J., "Officer Promotions: DOPMA Delivers Stability" Army Times, 10 June 1985.
8. Army Personnel Bulletin 3-85, Automated Officer Preference Statement, December 1985.
9. Department of the Army Regulation 614-100, Officer Assignment Policies, Details, and Transfers, 15 July 1984.
10. Department of the Army Pamphlet 600-3, Commissioned Officer Professional Development and Utilization, 25 July 1984.
11. Department of the Army Pamphlet 640-1, Officer's Guide to the Officer Record Brief, 15 January 1985.
12. Tice, J., "Officer Branch, Specialty Issue Resolved," Army Times, September 1985.
13. Jennings, M., "Specialty Code 15, Management in the Aviation Management Branch," Aviation Digest, April 1981.
14. Carothers, D., "Aviation Personnel Policy," Aviation Digest, July 1979.

15. "Implications of the New Army Aviator Career Pattern," Aviation Digest, June 1979.
16. Headquarters, 82nd Airborne Division, AFVC-AGP-O,  
Subject: Automation of the Officer Assignment  
Preference Statement, 18 September 1981.
17. "Most Couples in MAC Program Get Joint Posting," Army Times, 16 December 1985.
18. Carney, L., "Morrell Calls For Pinpoint Assignments," Army Times, 3 February 1986.
19. Tice, J., "Curious Mystique Sets Rank Apart," Army Times, 2 December 1985.
20. Army Personnel Bulletin 3-85, Regimental System Update, December 1985.

## BIBLIOGRAPHY

- Broome, R., An Analysis of Information Resource Management Within the Deputy Chief of Staff for Plans, U.S. Army Military Personnel Center, Master's Thesis, Naval Postgraduate School, CA, March 1985.
- Frazier, L., NPS Technical Note VM-14, THESIS9, Naval Postgraduate School, CA, November 1985.
- Nau, D., "Expert Computer Systems," COMPUTER, February 1983.
- NPS Technical Note VM-06, Script User's Guide for NPS, Naval Postgraduate School, CA, September 1984.
- Thesis Manual, Naval Postgraduate School, CA, November 1985.

# INITIAL DISTRIBUTION LIST

	No.	Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2	
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5000	2	
3. Computer Technology Programs, Code 37 Naval Postgraduate School Monterey, California 93943-5000	1	
4. Assistant Professor Daniel R. Dolk, Code 54dk Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5000	1	
5. Assistant Professor Tung Bui, Code 54bd Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5000	1	
6. Major Paul A. Stipek 33 Delawanda Drive Worcester, Massachusetts 01603	2	











Thesis  
S7264  
c.1

Stipek

CAESAR: Commis-  
sioned Assignments  
Executive Support for  
the US Army.

217586

9 FEB 88

32187

Thesis  
S7264  
c.1

Stipek

CAESAR: Commis-  
sioned Assignments  
Executive Support for  
the US Army.

217586



the  
CAESAR, Commissioned Assignments Execut



3 2768 000 66063 3

DUDLEY KNOX LIBRARY